software and hardware for the Bally Arcade

-a technical description

a dave nutting associates design

a division of bally manufacturing corporation

DAVE NUTTING ASSOCIATES, INCORPORATED ASSUMES NO RESPONSIBILITY FOR THE USE OF ANY CIRCUITRY OTHER THAN CIRCUITRY EMBODIED IN A DAVE NUTTING ASSOCIATES, INCORPORATED DESIGNED PRODUCT.

THIS DOCUMENT MUST BE RETURNED TO DAVE NUTTING ASSOCIATES, INCORPORATED BY REGISTERED MAIL WITHIN 5 DAYS UPON WRITTFN DEMAND.

TABLE OF CONTENTS - SOFTWARE

1		Home Video Game System
2		User Program Interface
5		System Routine Conventions
7		Inline Argument Mask Table Entry
8	INTPC	Begin Interpreting
9	XINTC	Exit Interpreter
1∅	RCALL	Call Assembly Language Subroutine
11	MCALL	Call Interpreter Subroutine
12	MJUMP	Interpreter Jump
13	MRET	Return From Interpretive Subroutines
14		Screen Handler
15	SETOUT	Set Display Ports
16	FILL	Fill A Contiguous Area With Constant
17	RECTAN	Paint A Rectangle
18		Screen Write Routines
19		Standard Calling Sequence
2∅		Pattern Representation
21	VWRITR	Write Relative From Vector
22	WRITR	Write Relative
23	WRITP	Write With Pattern Size Scare Up
24	WRIT	Write Pattern
25	WRITA	Write Absolute
26	SAVE	Save Area
27	RESTOR	Restore Area
28	VBLANK	Blank From Vector
29	BLANK	Blank Area
3∅	SCROLL	Scroll Window

31		Screen Alphanumeric Display Routines
34	DISNUM	Display BCD Number
35	DISTIM	Display Time
36	CHRDIS	Display Character
37	STRDIS	Display String
38		STRDIS Interpretation of Codes 64H to 7FH
39		Screen Vectoring - Vectoring Routines
42	VECT	Vector Object In Two Dimensions
43	VECTC	Vector A Co-ordinate
44	RELABS	Convert Relative Co-ordinates
45	RELAB1	Convert Relative Address To Absolute
46	COLSET	Set Color Registers
47	INCSCR	Increment Score And Compare To End Score
48	PAWS	Pause
49	KCTASC	Key Code to ASCII
5Ø	SENTRY	Sense Transition
53	DOIT	Respond To Input Transition
54	PIZBRK	Coffee Break, Black Out Screen, Wait For Key
55		Example
56		Interrupt - Music Processor
57		MUZCPU Instruction Set
58		Music Score Example
59	BMUSIC	Begin Playing Music
6Ø	EMUSIC	Stop Music
61	ACTINT	Active Interrupts
62	DECCTS	Decrement Counter/Timers
63	CTIMER	

64	STIMER	Decrement Timers
65	MOVE	Move Bytes
66	INDEXN	Index Nibble
67	STOREN	Store Nibble
68	INDEXW	Index Word
69	INDEXB	Index Byte
7Ø	SETB	Store Byte
71	SETW	Store Word
72		Cassette Conventions
75	GETPAR	Get Game Parameter
76	MENU	Display Menu And Branch On Selection
77	GETNUM	Get Number
79	MSKTD	Joystick Mask To Deltas
8Ø	RANGED	Ranged Random Number

TABLE OF CONTENTS - HARDWARE

81 Introduction

82 Memory Map

85 Screen Map

88 Color Mapping

89 Background Color

90 Vertical Blank

92 Interrupt Feedback

92 Interrupt Control Bits

93 Screen Interrupt

93 Light Pen Interrupt

94 Magic Register

95 Expand

96 Shifter

96 Flopper

98 Rotator

100 OR And XOR

100 Intercept

101 Player Input

103 Master Oscillator

104 Tones

104 Sound Block Transfer

106 Output Ports

107 Input Ports

1Ø9 Microcycler

111 Address Chip Description

114 Data Chip Description

117 I/O Chip Description

119 Music Processor

123 Custom Chip Timing

131 Video Timing

135 Electrical Specifications for Midway Custom Circuits

LIST OF ILLUSTRATIONS

6 Context Block Format

20 Pattern Representation

32 Option Byte

33 Alternate Font Descriptor

40 Vector Block

41 Vector Status Detail

41 Checks Mask Detail

44 Normal and Flopped Co-ordinate Systems

51 Keypad Mask Configuration

56 Voices Status Register

66 INDEXN

68 INDEXW

74 Cassette Map

78 Display Number Options

78 Character Display Options

83 Memory Map Low Resolution

84 Memory Map High Resolution

86 Screen Map Low Resolution

87 Screen Map High Resolution

91 Color Register Map

97 Shifter - Flopper

99 Rotator

102 Player Input

105 Audio Generator Block Diagram

106 Output Ports

1Ø7 Input Ports

1Ø8 System Block Diagram

11Ø Microcycler Block Diagram

113 Address Chip Block Diagram

116 Data Chip Block Diagram

118 I/O Chip Block Diagram

121 Master Oscillator

122 Tone Generators

124 Memory Write Without Extra Wait State

125 Memory Write With Video Wait State

126 Memory Read Without Extra Wait State

127 Memory Read With Video Wait State

128 I/O Read From Port 1ØH - 17H

129 I/O Read From Other Than Port'1ØH - 17H

13Ø I/O Write

132 Relationship Between 7M, Horiz Dr, Vert Dr, ΦG, $\overline{PX}$, and RAS

133 Relationship Between Horiz Dr, Horiz Blank, Horiz Sync, and Color Burst

134 Relationship Between Vertical Sync, Vertical Blank, and Vertical Drive

HOME VIDEO GAME SYSTEM

This documentation describes the Bally Home Video Game System. The description begins with a discussion of the major sub-sections of the system. Following this, each sub-section is presented in greater detail, with detailed particulars, such as calling sequences and resource use.

The major sub-sections of the system are:

The User Program Interface...which allows cassettes to reference the system routines through a standard interface. Includes an interpreter.

The Screen Handler...a complex of routines for creating screen images. Includes facilities for initialization, pattern, and character display, co-ordinate conversion, and object vectoring.

The Interrupt Processor...decrements timers, plays music, and produces sounds.

The Human Interface...reads keypad and control handles, inputs game selection and options.

Math Routines...a package of routines for manipulating floating BCD numbers.

USER PROGRAM INTERFACE

The User Program Interface (UPI) is a set of procedures and conventions, which are utilized by a cassette program to access the facilities provided by the home video game system. By adhering to these conventions a cassette program will be system independent, thus allowing improvements to be made to later versions of the system and on-board games, while maintaining upward compatability.

The basic rule for using the UPI is:

> With exception to the system DOPE vector, no cassette should ever address system ROM directly, or expect a given cell to always equal a certain value

The mechanism for calling a system routine is:

```
        RST     56
        DEFB    (routine # + option)
```

where routine number is an even number specifying which sub-routine to transfer to, symbolic identifiers, which are equated to routine numbers, are provided in HVGLIB.

Option is used to specify how arguments are being passed to the system routine. If option equals zero, the arguments are presumed to exist in CPU registers; if option equals 1, the arguments are taken to follow in line after the routine number/option byte. These arguments are loaded into the CPU registers automatically before the called routine is entered. The arguments required by each system routine are given in the routine's detail documentation.

The SYSTEM macro generates the sequence previously mentioned with option = Ø:

```
        SYSTEM  (routine #)
(example)
        SYSTEM  FILL
```

The SYSSUK macro generates the sequence previously mentioned with option = 1:

```
        SYSSUK  (routine #)
```

Frequently it is desirable to string several system routine calls together. If four or more calls follow in sequence, it is more efficient to utilize the interpreter. By using the interpreter we void the overhead of the RST 56 instruction by expecting a call index to immediately follow the call index or arguments used by the previous system routine.

Special call indexes are used to enter and exit interpretive mode:

Example:

```
        SYSTEM  INTPC              ;BEGIN INTERPRETING
        DO      FILL               ;DO FILL ROUTINE
        DEFW    NORMEM             ;STARTING AT TOP OF SCREEN
        DEFW    92*BYTEPL          ;CONTINUING FOR 92 LINES
        DEFB    Ø                  ;FILLED WITH ZEROES
        DO      CHRDIS             ;DO CHARACTER DISPLAY ROUTINE
        DEFB    Ø                  ;Y-AXIS POSITION OF CHARACTER
        DEFB    1Ø                 ;X-AXIS POSITION OF CHARACTER
        DEFB    8                  ;OPTIONS-PLOP,1Ø-ON,ØØ-OFF
        DEFB    'A'                ;CHARACTER TO BE DISPLAYED
        EXIT                       ;EXIT INTERPRETER
```

A block of call indexes have been set aside for the internal use of cassette programs. If a negative call index is encountered, the user's macro routine address table and argument table are utilized. The user is responsible for storing the addresses of these tables into dedicated system RAM cells.

All UPI routines are re-entrant.

Registers which are not defined as containing output parameters will not change.

SYSTEM ROUTINE CONVENTIONS

A system routine is coded like a conventional machine language subroutine, with the exception that output parameters are not passed through registers, but rather through the context block.

The context block is created by the RST 56 call. The user's register set (AF, BC, DE, HL, IX, IY) is pushed onto the stack. Register IY is set to point at this stack frame. Thus a copy of the input arguments exists in RAM which the system routine may refer to as needed. These arguments are also present in the registers when the system routine is entered; hence it is only necessary to refer to the context block when one has clobbered an input argument.

An output argument is returned to the caller by setting it in the context block. If a register was changed, but the associated cell in the context block was not, then the register will have it's old value on return. Thus a system routine is free to use any of the registers it needs without concern to saving and restoring. Moreover, the user can assume that no registers will change except those defined as returning an output argument.

The following illustration describes the context block and equates provided in HVGLIB for each field.

Four tables are used by the UPI in the control transfer process. The first two tables gives the routines starting address indexed via call number. The systems table is named SYSDPT. The user may extend this table by storing the address of his extended table into USERTB, USERTB+1. This address should point 128 bytes before the first entry.

The other two tables describe what in line arguments a call that specifies in line arguments should expect. This table gives a one-byte bitstring, also indexed via call number. The systems name is MRARGT, the user's address is in UMARGT, UMARGT must point 64 bytes ahead. Arguments must follow the call in a specified order.

Note that the context contains additional information not shown. This information exists both above and below the context. User programs should never use this information or even assume that it exists. The user should only address this area by using IY.

DISPLACEMENT	MEMORY CELL	EQUATE NAME
0	IY	CBIYL
1		CBIYH
2	IX	CBIXL
3		CBIXH
4	E	CBE
5	D	CBD
6	C	CBC
7	B	CBB
8	FLAGS	CBFLAG
9	A	CBA
A	L	CBL
B	H	CBH

CONTEXT BLOCK FORMAT

IN LINE ARGUMENT MASK TABLE ENTRY
TABLES MRARGT and UMARGT

If a bit corresponding to a register is set, the register is loaded. The order in which the arguments must appear is:

IX (L then H), E, D, C, B, A, L, H

If an argument isn't specified, it is omitted.

7	6	5	4	3	2	1	0
H	L	A	IX	B	C	D	E

UPI INTPC
BEGIN INTERPRETING

Calling Sequence:	SYSTEM INTPC
Aruguments:	None
Notes:	None
Description:	

See UPI description for explanation of interpreter

UPI XINTC

EXIT INTERPRETER

Calling Sequence:	EXIT
Arguments:	None

Description:

This code causes the interpreter to exit. Execution of machine instructions proceeds at the following location.

Restrictions:

This routine should only be called using the interpreter. A direct system call would produce unpredictable (and catastrophic) results.

UPI RCALL
CALL ASSEMBLY LANGUAGE SUBROUTINE

```
Calling Sequence:       DO      RCALL
                             or
                        DONT    RCALL
                        DEFW    (routine address)
Arguments:              HL=address of routine to call
```

Description:

RCALL may be used to call any assembly language subroutine from the interpreter. When the subroutine returns, interpretation proceeds at the next instruction.

When the assembly language routine receives control, HL will point at the routine's starting address; the other registers will contain their current values. Any changes made to the register set by the subroutine will not be passed along. To pass an output parameter, the subroutine must alter the context block, which is pointed at by IY.

Restrictions:

Assembler routine must not destroy IY.

```
Example:                DEFB    RCALL
                        DEFW    CLRAC
                            .
                            .
                            .
                CLRAC:  XOR     A
                        RET
```

UPI MCALL
CALL INTERPRETER SUBROUTINE

```
Calling Sequence:       SYSTEM  MCALL
                             or
                        SYSSUK  MCALL
                        DEFW    (routine address)
```

```
Arguments:              HL=Subroutine Address
```

Description:

MCALL is used to call an interpreter sequence as a subroutine. MCALL may be used from machine language as well as within an interpreted sequence. Calls may be nested infinitely, limited only by stack space (4 bytes per call)

To exit the interpreted subroutine, use MRET.

```
Example:                SYSSUK  MCALL
                        DEFW    ZAPALL
                             .
                             .
                             .
                ZAPALL: DO      FILL+1          ;DO FILL
                        DEFW    NORMEM
                        DEFW    ØFFFH
                        DEFB    Ø
                        DO      MRET            ;GO BACK TO CALLER
```

UPI MJUMP
INTERPRETER JUMP

Calling Sequence:

```
DO      MJUMP
    or
DONT    MJUMP
DEFW    (goto address)
```

Arguments: HL=Go to address

Description:

The current interpretive program counter is set to the contents of HL. The next instruction is fetched from that address.

Restrictions:

MJUMP must be called from the interpreter. The targets of all JUMPS must also be interpreted sequences.

Example:

```
          SYSTEM  INTPC           ;ENTER INTPC STEP
              .
              .
              .
          DO      MJUMP           ;JUMP TO END OF
          DEFW    END             ;INTPC STEP
              .
              .
              .
END:      DEFB    XINTC           ;EXIT INTERPRETER
```

UPI MRET

RETURN FROM INTERPRETIVE SUBROUTINES

Calling Sequence:	DO MRET
Arguments:	None

Description:

MRET causes execution to proceed at the instruction following the corresponding MCALL instruction. See MCALL for more information.

SCREEN HANDLER

The screen handler is a group of routines for generating frame buffer images. Included are entries for filling sections of the screen with constant data, the animation of figures, and the display of alphanumerics.

Many of these routines utilize the MAGIC functions provided by the custom chips. Since the status of these chips cannot be context-switched, many of these routines are not re-entrant. The user is responsible for preventing conflicts. This can be done by disabling interrupt, or implementing a semaphore.

SCREEN SETOUT

SET DISPLAY PORTS

Calling Sequence:

```
SYSTEM  SETOUT
     or
SYSSUK  SETOUT
DEFB    BLINE*2
DEFB    HORIZX/4
DEFB    INMOD
```

Arguments:

A=Data to output to INMOD (port EH)
B=Data to output to HORCB (port 9H)
D=Data to output to VERBL (port AH)

Output: None

Description: Outputs above data to ports
See hardware writeup for discussion of above ports.

SCREEN FILL

FILL A CONTIGUOUS AREA WITH CONSTANT

```
Calling Sequence:      SYSTEM  FILL
                            or
                       SYSSUK  FILL
                       DEFW    (first byte)
                       DEFW    (number of bytes)
                       DEFB    (data to fill with)
Arguments:             A =Data to fill with
                       BC=number of bytes to fill
                       DE=address to begin filling at
```

Description:

This routine sets the memory range DE to (DE+BC-1) to the specified constant.

Notes:

Fill can be used for screen clearing, or initialization of scratchpad RAM. It is re-entrant.

SCREEN RECTAN

PAINT A RECTANGLE

```
Calling Sequence:       SYSTEM  RECTAN
                            or
                        SYSSUK  RECTAN
                        DEFB    (X co-ordinate)
                        DEFB    (Y co-ordinate)
                        DEFB    (X size)
                        DEFB    (Y size)
                        DEFB    (color mask)
Arguments:              A =Color mask to write rectangle with
                        B =Y-size of rectangle in pixels
                        C =X-size of rectangle in pixels
                        D =Y co-ordinate for UL corner of rectangle
                        E =X co-ordinate for UL corner of rectangle
```

Description:

A rectangle of specified size of color mask is written at X,Y. RECTAN uses the MAGIC functions and is not re-entrant.

```
Example:                Put up a 3 X 4 rectangle of color 2 at 15,13.
                        DO      RECTAN
                        DEFB    15
                        DEFB    13
                        DEFB    3
                        DEFB    4
                        DEFB    1Ø1Ø1Ø1ØB
```

SCREEN WRITE ROUTINES

Virtually every video game involves the manipulation of animated figures. These figures are composed of patterns which are arbitrary pixel arrays. The write routines are used to transfer such patterns to the screen.

Five hierarchical levels of call are supported. The levels differ in the amount of preprocessing required by the user before calling. The highest level assumes that most of the parameters reside in a standard data structure, while the lowest level presumes that all arguments are in registers with all attendant transformations (such as relative-to-absolute conversion) already accomlished. The five levels are:

(1) Write from a Vector
(2) Write Relative
(3) Write Variable Pattern
(4) Write
(5) Write Absolute

Two transformations of the pattern may be performed prior to writing. They are FLOP and EXPAND. FLOP is mirroring the pattern on the X-axis. EXPAND is the translation of a 1-bit per pixel pattern into a 2-bit per pixel pattern. Since many patterns are only two-color, this allows for more efficient pattern storage. FLOP and EXPAND can both be done at the same time.

Three writing modes may be used. They are PLOP, OR, and XOR. PLOP is a conventional store into RAM. If OR is optioned, the data being written is ORed bit by bit with whatever was already there. Similarly, if XOR is set, the pattern is XORed with that beneath. Use of OR or XOR takes slightly longer since a read before write must be performed.

Note that ROTATE is not currently supported in software due to space considerations.

STANDARD CALLING SEQUENCE

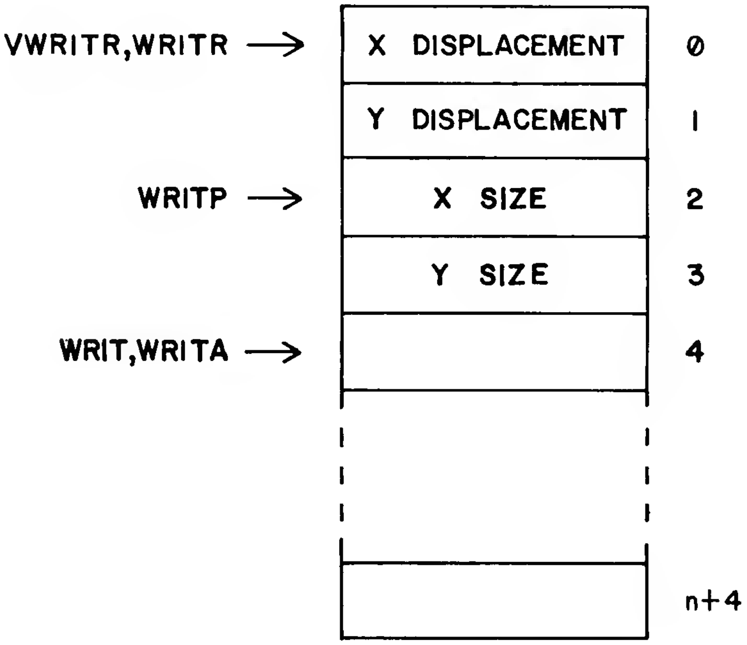
Every write routine uses a subset of the following argument/register assignment:

A = Magic Register
B = Y Pattern Size
C = X Pattern Size in Bytes
D = Y Co-ordinate (Ø - 1Ø1)
E = X Co-ordinate (Ø - 159)
HL = Pattern Address
IX = Vector Address

PATTERN REPRESENTATION

The higher the level of the write routine, the more ancillary information is stored with the pattern. The following diagram shows what each level expects. Any bytes of lower address than the pointer for a given level, need not be specified.

Use Restrictions:
None of the write routines are re-entrant due to Magic Register/Expander clobber.



SCREEN WRITE VWRITR

WRITE RELATIVE FROM VECTOR

Calling Sequence:	SYSTEM VWRITR
	or
	SYSSUK VWRITR
	DEFW (vector)
	DEFW (pattern)
Arguments:	HL=Pattern address
	IX=Vector Address
Output:	DE=Absolute address used
	A =Magic register used

Description:

The co-ordinates and magic register are loaded from the specified vector. (See vector routine document) The relative co-ordinates stored with the pattern are added to the co-ordinates from the vector. The pattern size is also taken from the pattern and writing proceeds.

Notes:

If expansion is to be done, the ON/OFF color must be set by the user before calling VWRITR.

SCREEN WRITE WRITR
WRITE RELATIVE

```
Calling Sequence:       SYSTEM  WRITR
                             or
                        SYSSUK  WRITR
                        DEFB    (X co-ordinate)
                        DEFB    (Y co-ordinate)
                        DEFB    (Magic Register)
                        DEFW    (Pattern address)
Arguments:              HL=Pattern address
                        A =Magic Register
                        D =Y co-ordinate
                        E =X co-ordinate
Output:                 DE=Screen Address Used
                        A = Magic Register Used
```

Description:
The relative co-ordinates stored with the pattern are added to the co-ordinates passed in DE. Pattern size is taken from the pattern.

Notes:
If expansion is to be done, the ON/OFF color must be set by the user before calling WRITR.

SCREEN WRITE WRITP

WRITE WITH PATTERN SIZE SCARE UP

Calling Sequence:	SYSTEM WRITP
	or
	SYSSUK WRITP
	DEFB (X co-ordinate)
	DEFB (Y co-ordinate)
	DEFB (Magic Register)
	DEFW (Pattern address)
Arguments:	HL=Pattern Address
	A =Magic Register
	D =Y co-ordinate
	E =X co-ordinate
Output:	DE=Screen Address Used
	A =Magic Register Used

Description:

The pattern size is taken from the pattern.

Notes:

User must worry about ON/OFF color if expansion is used.

SCREEN WRITE WRIT

WRITE PATTERN

Calling Sequence:	SYSTEM WRIT
	or
	SYSSUK WRIT
	DEFB (X co-ordinate)
	DEFB (Y co-ordinate)
	DEFB (X pattern size)
	DEFB (Y pattern size)
	DEFB (Magic Register)
	DEFW (Pattern address)
Arguments:	HL=Pattern Address
	A =Magic Register to use
	B =Y pattern size
	C =X pattern size
	D =Y co-ordinate
	E =X co-ordinate
Output:	DE=Absolute address used
	A =Magic Register used

Notes:

User must set ON/OFF color if using expansion.

SCREEN WRITE   WRITA

WRITE ABSOLUTE

```
Calling Sequence:       SYSTEM  WRITA
                             or
                        SYSSUK  WRITA
                        DEFW    (Absolute address)
                        DEFB    (X pattern size)
                        DEFB    (Y pattern size)
                        DEFB    (Magic Register)
                        DEFW    (Pattern address)
Arguments:              HL=Pattern Address
                        A =Magic Register
                        B =Y Pattern size
                        C =X Pattern size
                        DE=Absolute screen address of upper left-
                           hand corner of where to write
```

Notes:

This entry can be used for pattern writing to non-magic memory. The value in A is not output to (MAGIC); it is only interrogated to decide whether to FLOP or EXPAND.

SCREEN SAVE

SAVE AREA

Calling Sequence:

```
SYSTEM  SAVE
    or
SYSSUK  SAVE
DEFW    (save area)
DEFB    (X size)
DEFB    (Y size)
DEFW    (Screen address)
```

Arguments:

B =Y size of area to save

C =X size of area to save (in bytes)

DE=Address of save area

HL=Absolute address of upper left-hand corner of area to save

Description:

SAVE is used to preserve what is 'underneath' a moving pattern. SAVE copies the indicated area of the screen to the save area. The sizes of the area which was saved is preserved in the first two bytes of the save area.

The save area size must be greater than or equal to the X-size times the Y-size plus 2.

The save area may be MAGIC or non-MAGIC.

SCREEN RESTORE

RESTORE AREA

Calling Sequence:

```
SYSTEM  RESTOR
     or
SYSSUK  RESTOR
DEFW    (Save area)
DEFW    (Screen address)
```

Arguments:

DE=Save area to restore from

HL=Absolute address of upper left-hand corner of area to restore

Description:

RESTORE is the inverse of SAVE. The size of the area to restore is taken from the first two bytes of the save area.

SCREEN VBLANK
BLANK FROM VECTOR

```
Calling Sequence:       SYSTEM  VBLANK
                             or
                        SYSSUK  VBLANK
                        DEFW    (Vector address)
                        DEFB    (X size)
                        DEFB    (Y size)
Arguments:              D =Y size
                        E =X size (in bytes)
                        IX=Vector address
```

Description:
The BLANK bit in the vector status byte is tested. If it is not set, no blanking is done. If it is set, it is reset, then the old screen address is taken from the vector and blanking is done. If FLOPPED is specified by the Magic Register byte in the vector, a flopped blank is done. VBLANK always blanks to zero.

SCREEN BLANK

BLANK AREA

```
Calling Sequence:       SYSTEM  BLANK
                             or
                        SYSSUK  BLANK
                        DEFB    (X size)
                        DEFB    (Y size)
                        DEFB    (Blank to)
                        DEFW    (Blank address)
Arguments:              HL=Blank address (not MAGIC)
                        B =Data to blank to
                        D =Y size
                        E =X size
```

Description:

The specified area is blanked to whatever is passed in B.

SCREEN SCROLL
SCROLL WINDOW

```
Calling Sequence:        SYSTEM  SCROLL
                              or
                         SYSSUK  SCROLL
                         DEFW    (line increment)
                         DEFB    (# of bytes)
                         DEFB    (# of lines)
                         DEFW    (first byte)
Arguments:               B =Number of lines to scroll
                         C =Number of bytes on line to scroll
                         DE=Line increment
                         HL=First byte to scroll
```

Description:
This routine copies NBYTES from first line +INC to first line. Thus to scroll upward, HL points at the first line (which is overwritten) and the line increment would be positive. To scroll downward HL points at the last line and the line increment would be negative. The value in HL is an absolute address calculated by:
BASE OF SCREEN + #BYTES IN X OFFSET +(#lines offset*byte per line)

Note:
This routine can only be used to scroll one line at a time.

SCREEN ALPHANUMERIC
ALPHANUMERIC DISPLAY ROUTINES

HVGSYS provides several routines for the display of alphanumeric information. This section provides information which is common to all of the alphanumeric display routines.

The ASCII character code is used to represent all strings, with the following extensions:

> Characters with hex equivalents in the range 1 - 1F are interpreted as tabulation codes which cause the character display routines to skip over N character positions before writing the following characters.
>
> The characters 2ØH to 63H are displayed as 5 X 7 standard graphics with 3 pixels of horizontal spacing and 1 pixel of vertical spacing.
>
> The characters between 64H and 7FH are interpreted by STRDIS as control codes which cause the contents of registers C, DE, and IX to be changed to the value that follow the string. See table accompanying STRDIS.
>
> The characters between 8ØH and FFH are taken as references to a user supplied alternate character font.

The following argument/register combinations are used by all of the alphanumeric display routines.

Register C contains the options byte formatted as shown below.

ENLARGE FACTOR specifies if the character is to be enlarged in size. The table below defines the possible values for this parameter.

XOR/OR WRITE - all writes are performed through magic memory. Use of one of these options causes the character to be ORed/XORed with what was beneath it.

ON/OFF COLOR - all characters are stored one bit per pixel, but are written two bits per pixel by use of the expander. This field specifies the pixel values to translate the one bit per pixel representation into. For example, the value 11Ø1 specifies that the foreground color is 11, and the background color is Ø1.

OPTION BYTE

ENLARGE FACTOR	XOR WRITE	OR WRITE	ON COLOR	OFF COLOR

ENLARGE FACTOR	HOW MANY TIMES LARGER	ENLARGED SIZE OF SINGLE PIXEL
ØØ	1	1 X 1
Ø1	2	2 X 2
1Ø	4	4 X 4
11	8	8 X 8

D register contains the Y co-ordinate and the E register contains the X co-ordinate. These co-ordinates give the address of the upper left-hand corner where the first character will appear. Upon return, these registers are updated to give the address of the character to the right, (or below if no more space exists on the line). This simplifies the composition of complex messages.

IX register contains the Alternate Font Descriptor. It is required only if alternate font is referenced in call. Each character must be stored in one-bit per pixel format.

The small (3 X 5) character set is displayed using this facility. A word in the system DOPE vector points at a standard alternate font descriptor for this character set.

The format of the alternate font descriptor is shown below.

IX →	Ø	BASE CHARACTER	EQUAL TO FIRST CHARACTER IN TABLE
	1	X FRAME SIZE	CHARACTER SIZE IN BITS + X SPACING
	2	Y FRAME SIZE	CHARACTER SIZE IN BITS + Y SPACING
	3	X PATTERN SIZE	EACH CHARACTER TABLE ENTRY SHOULD BE OF SIZE X PATTERN*Y PATTERN SIZE
	4	Y PATTERN SIZE	
	5	CHARACTER TABLE ADDRESS	
	6		

SCREEN ALPHANUMERIC DISNUM
DISPLAY BCD NUMBER

Calling Sequence:	SYSTEM DISNUM
	or
	SYSSUK DISNUM
	DEFB (X)
	DEFB (Y)
	DEFB (options)
	DEFB (extended options)
	DEFW (number address)
Arguments:	B =Extended options
	C =Standard alphanumeric options byte
	DE=Standard X,Y co-ordinate
	HL=Address of BCD number
*NOT LOADED	IX=Optional character font descriptor
Outputs:	DE=Updated

Decription:

This routine displays the standard BCD codes Ø through 9. In addition, the codes AH through FH are also defined. The interpretation for these codes are:

A = * B = + C = '
D = - E = . F = /

If leading zero suppress is set, then instead of displaying a leading zero, a space is displayed. The first non-zero nibble encountered terminates leading zero suppression (including A - F). If the number is zero, a single zero is displayed.

If alternate font is set, the routine will display using codes between AAH and B9H (zero starting at BØH).

SCREEN ALPHANUMERIC   DISTIM

DISPLAY TIME

```
Calling Sequence:       SYSTEM  DISTIM
                             or
                        SYSSUK  DISTIM
                        DEFB    (X co-ordinate)
                        DEFB    (Y co-ordinate)
                        DEFB    (options)
Arguments:              DE=X,Y co-ordinates
                        X =Options  (see note below)
                        IX=Alternate Font Descriptor  (not loaded)
Outputs:                DE=Updated
```

Description:

This routine displays the system time (GTMINS,GTSECS) at the co-ordinates specified in the form MM:SS, where M=minutes, S=seconds. Seconds are optional.

Notes:

The small character set is used and one level of enlarge factor is permitted.

Options are the same as the alphanumeric display routine except that bit 7=1 to display colon and seconds; bit 7=Ø to suppress colon and seconds.

SCREEN ALPHANUMERIC CHRDIS
DISPLAY CHARACTER

```
Calling Sequence:       SYSTEM  CHRDIS
                             or
                        SYSSUK  CHRDIS
                        DEFB    (X co-ordinate)
                        DEFB    (Y co-ordinate)
                        DEFB    (options)
                        DEFB    (Character)
Arguments:              A =ASCII character to display
                        C =Standard options byte
                        DE=Standard Y,X co-ordinates to begin at
*NOT LOADED             IX=Optional alternate font descriptor address
Outputs:                DE=Updated to next frame
```

Description:

This is the basic charcter display promative. If tabulation is specified, the co-ordinates are updated but no actual writing occurs.

Notes:

Observe that IX is not loaded by the UPI SUCK facility. If alternate font is used, IX must be loaded with alternate font descriptor address.

Since this routine uses magic memory, it is not re-entrant.

SCREEN ALPHANUMERIC	STRDIS
DISPLAY STRING	
Calling Sequences:	SYSTEM STRDIS
	or
	SYSSUK STRDIS
	DEFB (X co-ordinate)
	DEFB (Y co-ordinate)
	DEFB (Options)
	DEFW (String)
Arguments:	HL=String address
	C =Standard Options
	DE=Standard Co-ordinates
*NOT LOADED	IX=Alternate Font Descriptor Address
Outputs:	DE=Updated to next frame

Description:

The string pointed at by HL is displayed as optioned. The string is terminated by a zero byte.

Notes:

IX is not loaded by SUCK. STRDIS is not re-entrant.

STRDIS INTERPRETATION OF CODES 64H to 7FH

STRDIS responds to the charcter codes between 64H and 7FH. These codes are taken to specify that certain registers in the context block are to be set to new values. This facility is useful for changing size, write mode, screen co-ordinates, or fonts, during a single STRDIS call.

The following table specifies which registers are loaded for a given code. The order in which the new register data follows the code, is also represented.

64H	C	72H	IX,D
65H	E,C	73H	IX,E,D
66H	D,C	74H	IX,C
67H	E,D,C	75H	IX,E,C
68H	NONE	76H	IX,D,C
69H	E	77H	IX,E,D,C
6AH	D	78H	IX
6BH	E,D	79H	IX,E
6CH	C	7AH	IX,D
6DH	E,C	7BH	IX,E,D
6EH	D,C	7CH	IX,C
6FH	E,D,C	7DH	IX,E,C
7ØH	IX	7EH	IX,D,C
71H	IX,E	7FH	IX,E,D,C

SCREEN VECTORING - VECTORING ROUTINES

Most games involve moving patterns. Most moving patterns move along a line. The home video game operating system provides the vectoring routines to facilitate programming such pattern motion.

The vectoring routines work with a memory array called a vector. Represented within this vector are the co-ordinates of an object, the velocities of the object, and the necessary status information to control the object. By periodically invoking the vectoring routine, this data is updated and can be used to direct the motion of a pattern.

More formally, a vectored object possesses an X and Y co-ordinate. Associated with these co-ordinates are velocities ΔX and ΔY, which are added to X and Y every time increment. Since the screen is finite, there also exists two upper and two lower limits X_{LU}, X_{LL}, Y_{LU}, and Y_{LL}, the attainment of which requires some response. The HVGSYS vectoring routine allows for two different responses to a limit attained. Either the sign of the delta is reversed or vectoring is stopped for this co-ordinate. This is specified by a flag byte. When attainment occurs, this fact is indicated by a status byte. Also the co-ordinate is set equal to the limit that was attained, preventing over-shoot.

Utilization of the vectoring routines involves a number of user responsibilities. The user must properly initialize certain fields in the vector array. He must increment the time base byte, and periodically call the vectoring routine. Status bits must be checked and writing must be done.

To insure high-accuracy, co-ordinates and deltas are double-precision. The assumed binary "decimal point" is between the high and low order byte.

The following diagrams explain the layout of the vector array and the attendant user responsibilities.

VECTOR BLOCK

BYTE	FUNCTION	HVGLIB NAME	
Ø	MAGIC REGISTER	VBMR	– DO NOT USE BIT 7
1	VECTOR STATUS	VBSTAT	
2	TIME BASE	VBTIMB	– INCREMENTED BY USER
3	Δ X	VBDXL	
4		VBDXH	
5	X	VBXL	
6		VBXH	
7	X CHECKS MASK	VBXCHK	
8	Δ Y	VBDYL	
9		VBDYH	
1Ø	Y	VBYL	
11		VBYH	
12	Y CHECKS MASK	VBYCHK	
13	OLD SCREEN ADDRESS	VBOAL	– MAINTAINED BY USER
14		VBOAH	

VECTOR STATUS DETAIL

ACTIVE VBSACT	BLANK VBBLNK	NOT USED					

ACTIVE Set by user to indicate that vector is active. The vectoring routines will do no processing if reset.

BLANK Must be initialized by user to reset state. Thereafter this bit is maintained by the VWRIT and VBLANK system routines.

CHECKS MASK DETAIL

NOT USED				LIMIT ATTAINED VBCLAT	NOT USED	REVERSE DELTA SIGN VBCREV	LIMIT CHECK VBCLMT

LIMIT CHECK Set by user to indicate that this co-ordinate is to be limit checked.

REVERSE DELTA Set by user to indicate that when this co-ordinate attains it's limit, the sign of the associated delta is to be reversed. This can be used to cause objects to 'bounce' off barriers.

LIMIT ATTAINED Set by system if the limit was attained this call. Otherwise it is reset. If the delta was not changed, either by Reverse Delta or user, this bit will stay set.

SCREEN VECTORING VECT

VECTOR OBJECT IN TWO DIMENSIONS

Calling Sequence:	SYSTEM VECT
	or
	SYSSUK VECT
	DEFW (Vector address)
	DEFW (Limit table)
Arguments:	HL=Limit table address
	IX=Vector address (points at VBMR)
Output:	C =Time base used
	Z =True, if it did not move

Description:

If the vector is inactive, control is returned immediately. Otherwise VECTC is called for X, then Y. The zero status is determined by comparing the new co-ordinate value with it's old value. If the high-order byte changed, then the object moved. Zero status set if object did not move, reset if object moved.

SCREEN VECTORING VECTC
VECTOR A CO-ORDINATE

```
Calling Sequence:     SYSTEM  VECTC
                          or
                      SYSSUK  VECTC
                      DEFW    (co-ordinate address)
                      DEFW    (Limit table)
Arguments:            IX=Pointer to low-order byte of delta for co-ordinate
                      HL=Limits table for this co-ordinate (if required)
                      C =Time base to use
```

Description:
This routine operates on the subset of the vector array associated with a single co-ordinate. This subset consists of the delta co-ordinate and checks mask. This entry is provided so special vectoring schemes may be implemented such as 1 dimensional or 3 dimensional vectoring.

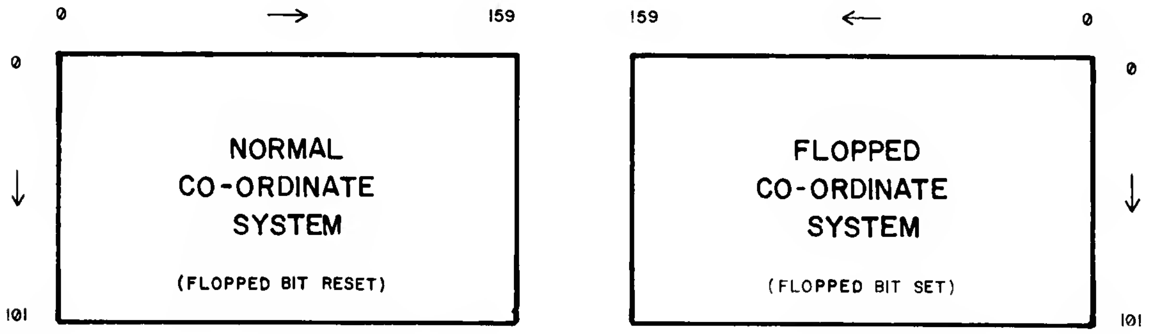
This entry adds the delta to the co-ordinate time base times. It then performs the limit checks for the co-ordinate if optioned.

Note that this entry does not interrogate or alter any bytes in the vector array outside of the defined subset. Hence the active bit isn't checked.

SCREEN RELABS
CONVERT RELATIVE CO-ORDINATES TO ABSOLUTE MAGIC ADDRESS AND
SET UP MAGIC REGISTER

Calling Sequence:	SYSTEM RELABS
	or
	SYSSUK RELABS
	DEFB (Magic register value)
Arguments:	A =Magic register value to set
	D =Y co-ordinate
	E =X co-ordinate
Output:	A =Magic register value, with proper shift amount set
	DE=Absolute memory address (MAGIC)

Description:
The low-order two bits of the X co-ordinate are inserted into the magic register value bitstring. The absolute memory address corresponding to the co-ordinate is computed, taking into consideration the value of the flopped bit. The co-ordinate systems used are shown below.



SCREEN RELAB1

CONVERT RELATIVE ADDRESS TO ABSOLUTE NORMAL ADDRESS

Calling Sequence:	SYSTEM RELAB1
	or
	SYSSUK RELAB1
	DEFB (Magic register value)
Arguments:	A =Magic register value to combine with shift amount
	D =Y co-ordinate
	E =X co-ordinate
Output:	A =Combined magic register value
	DE=Absolute normal address (not magic)

Description:

This routine is identical to RELABS except that a non-magic address is returned and the hardware magic register is not set. The flopped bit is interrogated and the flopped co-ordinate system is used, if optioned.

SCREEN COLSET

SET COLOR REGISTERS

```
Calling Sequence:       SYSTEM  COLSET
                            or
                        SYSSUK  COLSET
                        DEFW    (Address of color list)
Inputs:                 HL=Color list laid out
                      COL3L=first to
                      COLOR last  ie: COLOR would be at a higher
                                      address than COL3L
```

Description:

This routine sets color registers and saves address of colors for use by PIZBRK and BLAKOUT for color restoration.

HUMAN INCSCR

INCREMENT SCORE AND COMPARE TO END SCORE

```
Calling Sequence:       SYSTEM  INCSCR
                             or
                        SYSSUK  INCSCR
                        DEFW    (address of score)
Arguments:              HL=Address of score (must be 3 bytes long)
Output:                 Score incremented and optionally game over bit set
```

Description:

The 3 byte score pointed at by HL (BCD with low-order byte at lowest address) is incremented (by 1) and compared to the end score (ENDSCR). If the end score bit (GSBSCR) was set in the game status byte (GAMSTB) and end score has been reached, then the game over bit (GSBEND) is set in the game status byte.

HUMAN PAWS

PAUSE

Calling Sequence: SYSTEM PAWS
or
SYSSUK PAWS
DEFB (number of interrupts)

Arguments: B=Number of interrupts to wait

Description:
This routine provides for a pause for certain number of interrupts. If used with ACT INT, 6Ø will be a 1-second pause. This routine does an EI upon entry and assumes interrupts will occur.

HUMAN KEYBOARD KCTASC
KEY CODE TO ASCII

Calling Sequence:	SYSTEM KCTASC
Arguments:	B=Key code (not loaded)
Output:	A=ASCII equivalent of keycode
Description:	This routine does a table look-up

KEYCODE	NAME	GRAPHIC	HEX VALUE
1	Clear	C	43
2	Up Arrow	↑	5E
3	Down Arrow	↓	5C
4	Percent	%	25
5	Recall	MR	52
6	Store	MS	53
7	Change sign	CH	3B
8	Divide	÷	2F
9	7	7	37
10	8	8	38
11	9	9	39
12	Times	X	2A
13	4	4	34
14	5	5	35
15	6	6	36
16	Minus	-	2D
17	1	1	31
18	2	2	32
19	3	3	33
20	Plus	+	2B
21	Clear Entry	CE	26
22	Ø	Ø	3Ø
23	Decimal point	.	2E
24	Equals	=	3D

HUMAN CONTROLS & KEYPAD   SENTRY
SENSE TRANSITION

```
Calling Sequence:       SYSTEM  SENTRY
                             or
                        SYSSUK  SENTRY
                        DEFW    (Key mask address)
Arguments:              DE=Keypad mask table
```

Description:
SENTRY checks for changes in the potentiometers (pots), control handles, triggers, keypad, semiphores and counter/timers. It also takes care of blackout. Blackout is the automatic blacking-out of the screen after 255 seconds without a change. If SENTRY isn't called then the game will not black out.

SENTRY checks if TIMOUT equals Ø on entry and if zero, it goes to PIZBRK. If a key has gone down or a control handle changed, then TIMOUT is set to FFH.

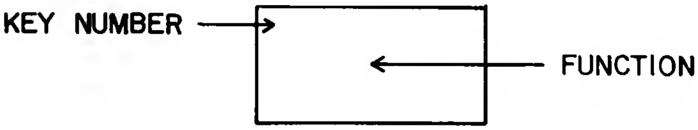
HL should point at a keypad mask. The keypad consists of 6 rows by 4 columns.

```
Example mask of         DEFB    Ø111ØØB
just Ø - 9              DEFB    1111ØØB
                        DEFB    Ø111ØØB
                        DEFB    ØØØØØØB
```

See diagram on following page.

1	2	3	4	MASK BIT NUMBER
1 C	2 ↑	3 ↓	4 %	0
5 MR	6 MS	7 CH	8 ÷	1
9 7	10 8	11 9	12 X	2
13 4	14 5	15 6	16 −	3
17 1	18 2	19 3	20 +	4
21 CE	22 0	23 .	24 =	5

MASK BYTE NUMBER



Output: A=Return code
B=Extended code

PRIORITY	A=	MEANING
	SNUL	Nothing changed
1	SCTØ	Counter/timer Ø decremented to Ø
	to	
1	SCT7	Counter/timer 7 decremented to Ø
2	SFØ	SEMI4S bit Ø was 1
	to	
2	SF7	SEMI4S bit 7 was 1
4	SSEC	1 second has elapsed since the last SSEC
5	SKYU	Keypad went from down to up B=Ø
5	SKYD	Key is down B=key number
3	SPØ	Pot Ø changed B=new value
	to	
3	SP3	Pot 3 changed B=new value
6	SJØ	Joystick Ø changed B=new value
	to	
6	SJ3	Joystick 3 changed B=new value
6	STØ	Trigger Ø changed B=new value
	to	
6	ST3	Trigger 3 changed B=new value

Notes:
The potentiometers (pots) are debounced. New trigger value=Trigger off (Ø) or trigger on (1ØH). When switches are actuated simultaneously the order of return is: SCT7 to SCTØ, SF7 to SFØ, SPØ to SP3, SSEC, SKYU, SKYD, SJØ, STØ, SJ1, ST1, SJ2, ST2, SJ3, ST3.

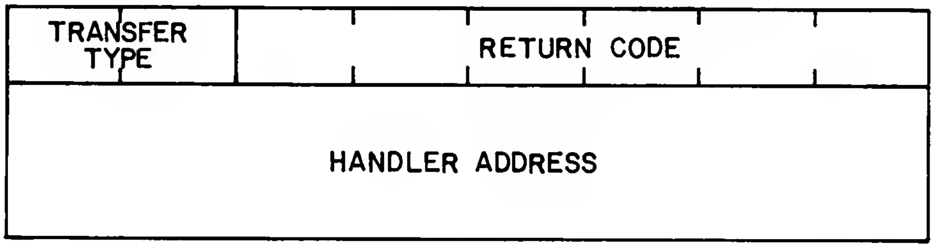
HUMAN CONTROL DOIT
RESPOND TO INPUT TRANSITION

Calling Sequence:	SYSTEM DOIT or SYSSUK DOIT DEFW (Do table)

Arguments:	A =SENTRY return code
	B =Extended return code
	HL=Do table address

Description:

The SENTRY return code is used to search the DOTABLE. If the transition is present in DOTABLE, then control is transferred to the associated handling routine. The handling routine may be MACRO or machine instructions. The routine receives registers as they are on DOIT entry. If no transition is found, execution continues at the first instruction following call. The DOTABLE is a linear list composed of 3 bytes entries, 1 entry per SENTRY return code.



Where transfer type designates how handler address is to be transferred to. The codes are: ØØ=JMP to machine language routine and pop context; Ø1=RCALL machine language routine in current context; 1Ø=MCALL interpreter routine in current context. Mode Ø1 and 1Ø expect the returned-to point to be interpretive, mode Ø expects it to be machine instructions.

End of list is indicated by a terminator byte which is greater than or equal to CØH.

HUMAN CONTROL PIZBRK

"COFFEE BREAK" BLACK OUT SCREEN AND WAIT FOR KEY

Calling Sequence:	SYSTEM PIZBRK
	or
	SYSSUK PIZBRK
Input:	NONE
Output:	NONE

Description:

This routine blacks out the screen and waits for either a key press or a trigger or a joystick change.

This function should be called whenever a "hold until further notice" is needed.

All keys on the keypad are enabled. Interrupts are disabled on entry and enabled on exit. It is a good idea to reset any 60th of a second timers on exiting PIZBRK.

HUMAN CONTROLS   EXAMPLE

This routine echoes number keys and takes a coffee break on trigger Ø being pulled. Assumes SP is set and screen erases.

```
        SYSTEM  INTPC
LOOP:   DO      SENTRY
        DEFW    NUMBAS
        DO      DOIT
        DEFW    DTAB
        DO      MJUMP
        DEFW    LOOP

NUMBAS: DEFB    Ø111ØØB         ;NUMBER KEYS ONLY
        DEFB    111ØØB
        DEFB    Ø111ØØB
        DEFB    Ø

DTAB:   MC      SKYD,SHOW       ;ON KEY DOWN MACRO CALL
        MC      STØ,PBREAK+END  ;ON TO MACRO CALL

SHOW:   DO      KCTASC          ;CONVERT TO ASCII
        DO      SUCK
        DEFB    ØØØØØ111B       ;X,Y=Ø=DE
        DEFB    11ØØ11ØØB       ;OPTIONS=C
        DONT    CHRDIS          ;DISPLAY CHAR
        MRET                    ;BACK TO LOOP

PBREAK: DO      PIZBRK          ;COFFEE BREAK
        DO      MRET            ;BACK TO LOOP
```

INTERRUPT MUSIC PROCESSOR

The music processor can be thought of as an independent CPU handling all output to the music/noise ports. The MUZCPU has 4 registers:

MPC: Like all program counters, points to the next data byte to fetch.

MSP: Like a stack pointer, points to return addresses on the stack.

DURATION: Is loaded at the start of a note and then decremented every 60th of a second

VOICES: Is a status register. It tells which voices (tones) to load with what data.

The voices status register is shown below. Execution proceeds right-to-left. Make sure that you always have at least one PC incrementing bit or load on.

INC PC	OUT TONE A	INC PC	OUT TONE B	INC PC	OUT TONE C	OUT VIBRA	OUT VOLN

MUZCPU INSTRUCTION SET

# OF BYTES	MNEMONIC	COMMENT
2	VOICES,(data)	;VOICES=(data)
2	MASTER,(data)	;TONEØ=(data)
3	CALL,(address)	;(SP)=(PC+3) PC=address
1	RET	;PC=(SP++)
3	JP,(address)	;PC=address
2	NOTE1	;Duration, note or data (D1)
3	NOTE2	;Duration, D1,D2
4	NOTE3	;Duration, D1,D2,D3
5	NOTE4	;Duration, D1,D2,D3,D4
6	NOTE5	;Duration, D1,D2,D3,D4,D5
2	REST	;Duration in 6Øths of a second ;Pauses silently (except legato)
1	QUIET	;Stops music and sets volume=Ø
2	OUTPUT	;Port #, Data
9	OUTPUT	;SNDBX,DATA1Ø,D11,D12,D13,D14,D15,D16,D17
3	VOLUME	;(VOLAB),(VOLMC) sets volume for notes
1	PUSHN	;Push # between 1-16 onto the stack
1	CREL	;Call relative to next instruction
3	DSJNZ	;decrement stack top and jump ;if not Ø, else pop stack
1	LEGSTA	;flips between STACATO and LEGATO modes ;STACATO is clipped 1/60th before the ;end of each note ;LEGATO allows one note to run into ;the next

Note: All durations are limited to a maximum of 127

MUSIC SCORE EXAMPLE

```
VOICES  11Ø1Ø1ØØB        ;ABC=Data 1
MASTER  ØA1H             ;ABC=½
VOLUME  88H,Ø8H
NOTE1   12,A1
NOTE1   12,C2
NOTE1   24,E2
NOTE1   12,C2
NOTE1   12,E2
REST    6
VOICES  1111Ø11ØB        ;Suck in Vibrato, AB and C bytes
NOTE3   12,14,A2,E2
QUIET
```

INTERRUPTS   MUSIC   BMUSIC

BEGIN PLAYING MUSIC

```
Calling Sequence:       SYSTEM  BMUSIC
                             or
                        SYSSUK  BMUSIC
                        DEFW    (Music stack)
                        DEFB    (voices byte)
                        DEFW    (Score)
Arguments:              A =Voices to start with
                        HL=Music PC (Score)
                        IX=Music SP
```

Description:

Quiets any previous music, then interprets "score". See music processor for more information.

INTERRUPTS MUSIC EMUSIC

STOP MUSIC

Calling Sequence:	SYSTEM EMUSIC or SYSSUK EMUSIC
Arguments:	NONE
Outputs:	NONE

Description:

Outputs Ø to volume ports and halts music processor.

INTERRUPTS   ACTINT

ACTIVE INTERRUPTS

Calling Sequence:	SYSTEM ACTINT or SYSSUK ACTINT
Input:	NONE
Output:	NONE
Function:	Sets IM=2, INLIN=2ØØ, sets I reg + INFBK Calls TIMEX and TIMEY Enables interrupts

Description:

Once ACTINT is called, it provides interrupt service completely automatically. It runs the seconds timer, the game timer, the music processor, and black-out timers, plus CTØ, CT1, CT2, CT3. Functions as 6Øth of a second timers.

INTERRUPTS   TIMERS   DECCTS

DECREMENT COUNTER/TIMERS

Calling Sequence:

```
SYSTEM  DECCTS
     or
SYSSUK  DECCTS
DEFB    (Mask)
```

Input: C=Mask indicative which counters to decrement.

Output: Sentry will notify the program.

Description:

Decrements counter if they are non-zero. If any go from 1 to Ø, sentry is notified.

INTERRUPTS   TIMERS   CTIMER

Calling Dequence: CALL CTIMER

Input: HL=Address of custom time base
B =Value to load into time base 1 to Ø transition
C =CT mask as in DECCTS

Description:

HL is loaded and decremented. If it is not = Ø, then a return is executed. Else, HL is loaded with B and DECCTS is called.

Registers HL, DE, BC, and AF are undefined upon exit.

INTERRUPTS   TIMERS   STIMER

DECREMENT TIMERS

Calling Sequence:

```
PUSH    AF
PUSH    BC
PUSH    DE
PUSH    HL
CALL    STIMER
POP     HL
POP     DE
POP     BC
POP     AF
```

Input: NONE

Description: STIMER keeps track of game time. If it hits Ø, then the GSBEND bit in the game status byte is set.

Uses: AF, BC, DE, HL

Calls: Music processor on note (duration) expiration.

Note: Sets bit 7 of key sex to 1 on every second.

MOVE   MOVE BYTES

Calling Sequence:	SYSTEM MOVE
	or
	SYSSUK MOVE
	DEFW (Destination)
	DEFW (Number of bytes)
	DEFW (Source)
Arguments:	DE=Destination address
	HL=Source address
	BC=Number of bytes to transfer
Description:	MOVE uses LDIR to copy bytes from source to destination.

INDEXN INDEX NIBBLE

Calling Dequence:	SYSTEM INDEXN or SYSSUK INDEXN DEFW (Base Address)
Arguemnts:	C =Nibble displacement (Ø - 255) HL=Base address of table
Output:	A =Nibble value

Description:
INDEXN is used to look up a given nibble in a linear list. The indexing works like:

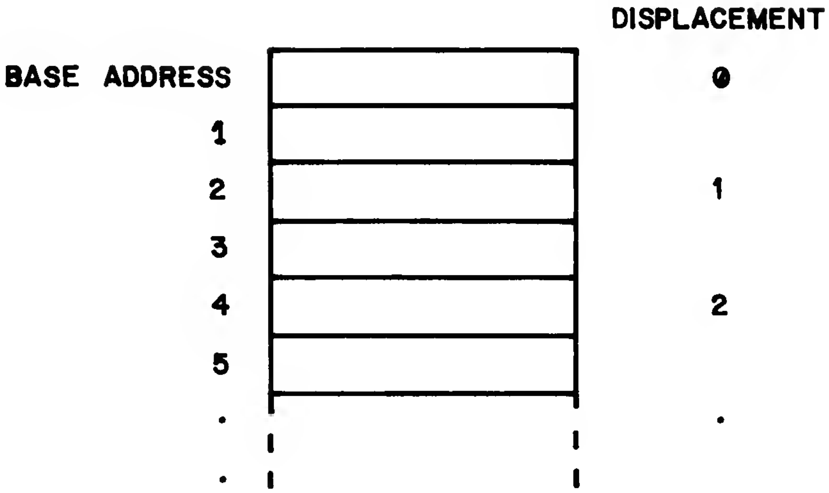
BASE ADDRESS	1	0
1	3	2
2	5	4
3	7	6
.		
.		

STOREN STORE NIBBLE

Calling Dequence:	SYSTEM STOREN	
	or	
	SYSSUK STOREN	
	DEFW (Base address)	
Arguments:	C =Nibble displacement	*NOT LOADED
	HL=Base address	
	A =Nibble value to store	*NOT LOADED
Description:	STOREN is the inverse of INDEXN. STOREN works as with INDEXN.	

INDEXW INDEX WORD

```
Calling Sequence:       SYSTEM  INDEXW
                             or
                        SYSSUK  INDEXW
                        DEFW    (Base address)
Arguments:              A =Displacement (Ø - 255)       *NOT LOADED
                        HL=Base address of table
Output:                 DE=Entry looked up
                        HL=Address of entry looked up
Description:            Indexing looks like:
```



INDEXB   INDEX BYTE

Calling Sequence:	SYSTEM INDEXB
	or
	SYSSUK INDEXB
	DEFW (Base address)
Arguments:	A =Displacement (Ø - 255)
	HL=Base address of table
Output:	A =Entry looked up
	HL=Address of entry looked up

Notes:

INDEXB returns the byte at address

(Base address) + (Displacement)

SETB STORE BYTE

Calling Sequence: SYSTEM SETB

```
Calling Sequence:     SYSTEM  SETB
                         or
                      SYSSUK  SETB
                      DEFB    (Value to store)
                      DEFW    (Address)
Arguments:            A =Byte value to store
                      HL=Address to be set
Description:          Stores an 8-bit value at a specified address.
```

SETW STORE WORD

Calling Sequence: SYSTEM SETW

or

SYSSUK SETW

DEFW (Value to store)

DEFW (Address)

Arguments: DE=Word value to store

HL=Address to be set

Description: Stores a 16-bit value at a specified address.

CASSETTE CONVENTIONS

Two types of cassettes may be used with the Bally Professional Arcade. The first type, called an autostart cassette, is entered immediately after reset. The only initialization that is performed before entry is the set-up of the stack pointer to point just below system RAM and the establishment of "consumer mode" in the custom chips. RAM is not altered in this mode.

The second type, called a standard cassette, is entered after a game selection process is completed. Considerably more initialization is done by the system before control transfer.

1) System RAM is cleared to Ø
2) The ACTINT interrupt routine is enabled
3) The MENU colors are set in the left color map
4) Vertical blank is set at line 96, horizontal boundary at 41, and interrupt mode at 8.
5) The screen displays the menu frame.
6) The shifter is cleared.

An autostart cassette is indicated by a jump instruction (opcode C3H) at location 2ØØØH. This jump instruction should branch to the starting address of the cassette.

A standard cassette is indicated by a sentinel byte of 55H at location 2ØØØH. Following this byte is the first node of the cassette's menu data structure. This data structure gives the name and starting address of each program in the cassette. (See MENU)

When the user has selected a cassette game, control is transferred to the starting address with the address of the program name string in the registers. The cassette program will use the GETPAR system routine to prompt for game parameters such as score to play to, game time limit or number of layers.

The cassette has access to the six unused restart instructions. See the following cassette diagram for the transfer vectors.

BYTE		
2000	0 1 0 1 0 1 0 1	SENTINEL
1	NEXT MENU NODE	MENU NODE FOR FIRST GAME ON CASSETTE
2		
3	STRING ADDRESS FOR FIRST GAME	
4		
5	START ADDRESS FOR FIRST GAME	
6		
7	RST 8 JUMP VECTOR	THESE CELLS MAY BE USED FOR PROGRAM IF THE ASSOCIATED RST OR HOOK IS NOT USED
8		
9		
A	RST 16	
B		
C		
D	RST 24	
E		
F		
2010	RST 32	
1		
2		
3	RST 40	
4		
5		
6	RST 48	
7		
8		
9	SENTRY HOOK TRANSFER VECTOR USED FOR DEMO PROGRAMS	
A		
B		

HUMAN GETPAR

GET GAME PARAMETER

```
Calling Sequence:       SYSTEM  GETPAR
                           or
                        SYSSUK  GETPAR
                        DEFW    (Prompt)
                        DEFB    (Digits)
                        DEFW    (Parameter)
Arguments:              A =Number of digits to get
                        BC=Address of prompt string
                        DE=Title string address          *NOT LOADED
                        HL=Address of parameter to get
```

Description:

A menu frame is created displaying the title passed in DE at the top. The message "ENTER" is displayed in the center of the screen followed by the prompt string. GETNUM is entered with feedback specified in 2X enlarged characters. After entry is complete, GETPAR pauses for ¼ second to allow user to see his entry and then returns.

Notes:

See entry conditions and resource requirements for menu.
Prompt string example: "# OF PLAYERS"
The title string address (DE) is usually the title returned from MENU.
The address of parameter to get (HL), HL points at the low-order byte of BCD number in RAM.

HUMAN MENU

DISPLAY MENU AND BRANCH ON SELECTION

Calling Sequence:	SYSTEM MENU or SYSSUK MENU DEFW (Title) DEFW (List)
Arguments:	DE=Address of menu title string HL=Address of menu list
Output:	DE=String address of selection mode

Description:

The title is displayed at the top of the screen. Each entry in the menu list is then displayed with a preceding number supplied by MENU. GETNUM is called to get the selection number. The menu list is searched for the selected node and it is jumped to.

Notes:

A maximum of eight entries may appear.

On entry, MENU expects interrupts to be enabled, colors and boundaries to be set up. MENU uses 96 lines of screen, creams the alternate set, and requires three levels of context. MENU calls SENTRY and thus 'eats' all irrelevant transitions.

NEXT	ADDRESS OF NEXT NODE ON LIST ZERO IF THIS NODE IS LAST
STRING	ADDRESS OF NAME OF THIS SELECTION THIS IS WHAT IS PASSED IN DE
GO TO	WHERE TO BRANCH TO IF THIS SELECTION IS SELECTED

HUMAN GETNUM

GET NUMBER

```
Calling Sequence:       SYSTEM  GETNUM
                             or
                        SYSSUK  GETNUM
                        DEFB    (X address)
                        DEFB    (Y address)
                        DEFB    (CHRDIS options)
                        DEFB    (DISNUM options)
                        DEFW    (Number address)
Arguments:              B =Display number routine options
                        C =Character display routine options
                        DE=Y,X co-ordinate for feedback
                        HL=Address of where to put entered number
```

Description:

This routine inputs a number from either the keypad or the pot on control handle of player one. Keypad entry has priority. The routine exits when the specified number of digits were entered or = is pressed on the keypad.

Pot entry is enabled by pressing the trigger. The current pot value is then shown. Twist the pot until the number you want is shown. Then press the trigger again to complete entry. The pot can only enter 1 or 2 digits. If a group of numbers is being entered, the user must enable entry for each new number.

DISPLAY NUMBER OPTIONS

ZERO SUPP	ALT FONT	NUMBER OF DIGITS TO DISPLAY/ACCEPT

CHARACTER DISPLAY OPTIONS

ENLARGE FACTOR	XOR	OR	ON COLOR	OFF COLOR

HUMAN MSKTD
JOYSTICK MASK TO DELTAS

Calling Sequence:

```
SYSTEM  MSKTD
     or
SYSSUK  MSKTD
DEFW    (X Delta)
DEFB    (Flop-flag)
DEFW    (Y Delta)
```

Arguments:
B =Joystick mask *NOT LOADED
C =Flop flag
DE=X positive delta
HL=Y positive delta

Output:
DE=X Delta
HL=Y Delta

Description:

This routine uses the joystick mask and flop flag to conditionally modify the passed deltas. If negative direction is indicated, the delta is 2's complemented; if no direction is indicated, Ø is returned.

Note: B is not sucked.

MATH   RANGED

RANGED RANDOM NUMBER

```
Calling Sequence:        SYSTEM  RANGED
                              or
                         SYSSUK  RANGED
                         DEFB    (N)
Arguments:               A=N where Ø is less than or equal to a random
                             numner less than N
                             (ie: for a random number of Ø,1,or 2, N=3)
Output:                  A=Random Number
```

Notes;

If N is a power of 2, it is considerably faster to use N=Ø which causes an 8-bit value to be returned without ranging. Use an AND instruction to range it yourself.

This routine uses a polynomial shift register RANSHT in system RAM. RANGED is called in GETNUM while waiting for game selection/parameter entry. Thus each execution of a program will receive different random numbers. For 'predictable' random numbers, alter RANSHT yourself after parameter acceptance.

INTRODUCTION

The Bally Professional Arcade is a full-color video game system based on the mass-ram-buffer technique. A mass-ram-buffer system is one in which one or more bits of RAM are used to define the color and intensity of a pixel on the screen. The picture on the screen is defined by the contents of RAM and can easily be changed by modifying RAM.

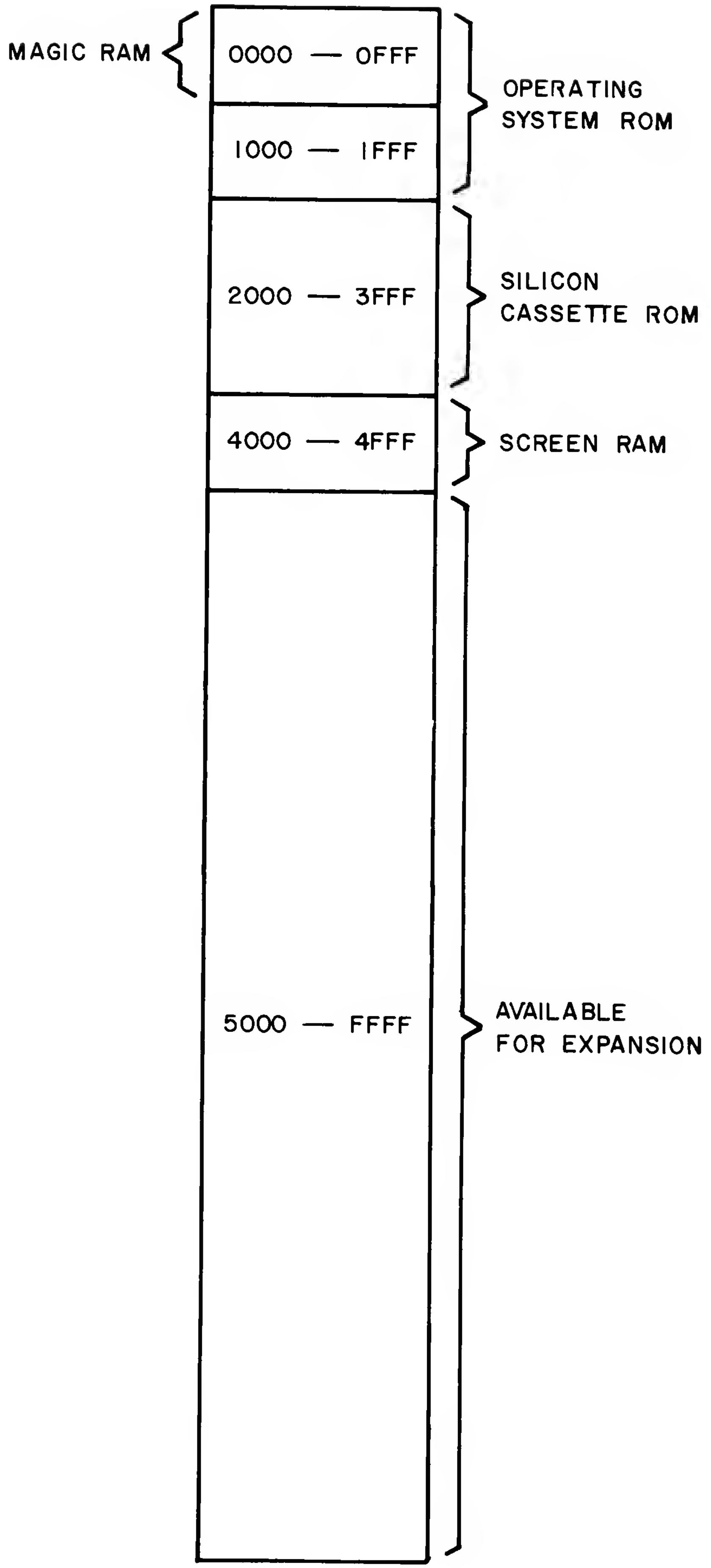
The system uses a Z-80 Microprocessor as it's main control unit. The system ROM has software for four games: Gunfight, Checkmate, Scribbling, and Calculator. Additional ROM can be accessed through the silicon cassette connector. Three custom chips are used for the video interface, special video processing functions, keyboard and control handle interface, and audio generation.

The system exists in both high-resolution and low-resolution models. The three custom chips can operate in either mode. The mode of operation is determined by bit Ø of output port 8H. It must be set to Ø for low-resolution and 1 for high-resolution. This bit is not set to Ø at power up and must be set by software before any RAM operations can be performed.

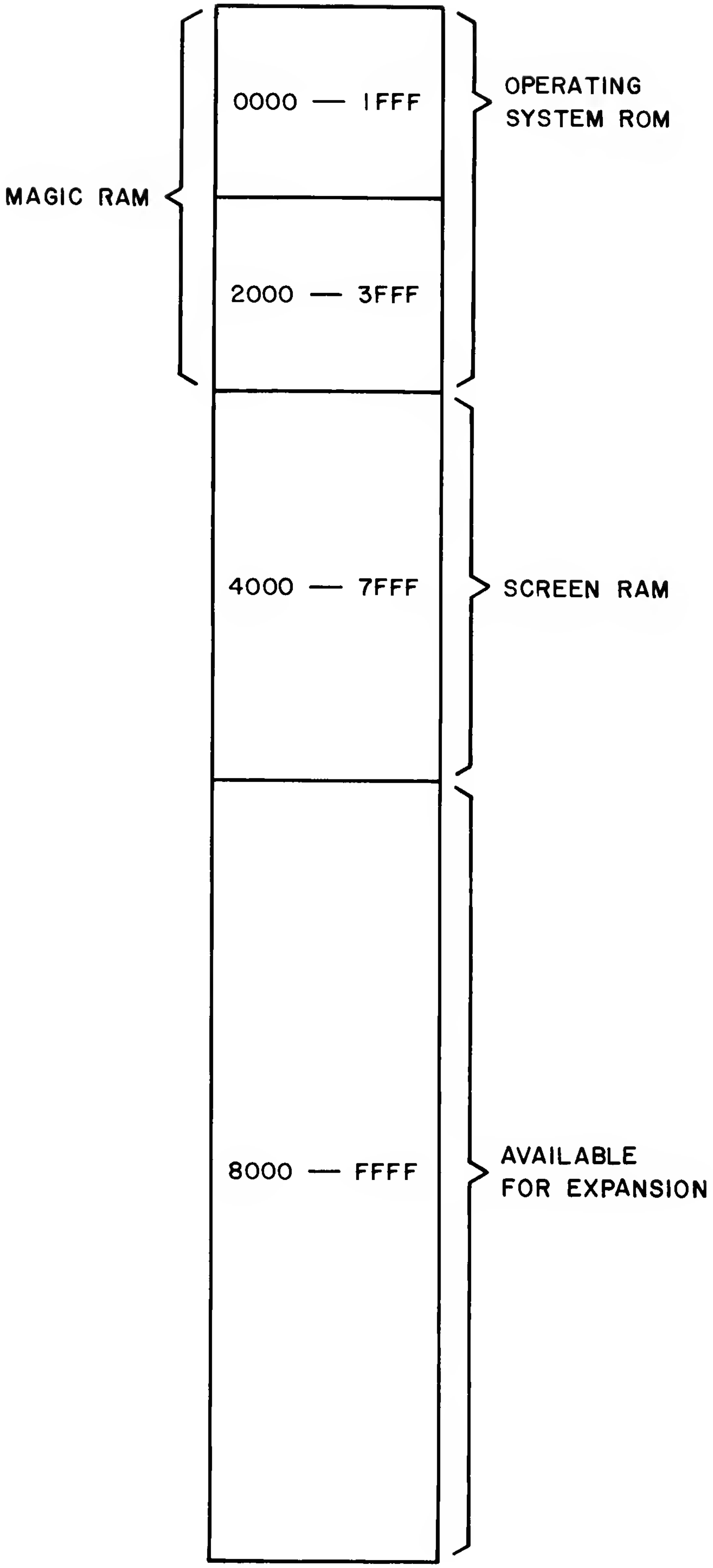
MEMORY MAP

In both the low and high resolution models, the operating system ROM is in the first 8K of memory space. The silicon cassette ROM is in the space from 8K to 16K. The standard screen RAM begins at 16K. In the low-resolution unit, standard screen RAM is 4K bytes; in the high-resolution unit it is 16K bytes. Magic screen RAM begins at location Ø. It is the same size as standard screen RAM. All memory above 32K is available for expansion. In the low-resolution unit, memory space 20K - 32K is available for expansion.

When data is read from a memory location between Ø and 16K the data comes from the ROM. When data is written in a memory location (X) between Ø and 16K, the system actually writes a modified from of the data in location X+16K. The modification is performed by the magic system in the Data Chip and Address Chip. Thus the RAM from 0 to 16K is called Magic Memory.



MEMORY MAP LOW RESOLUTION



MEMORY MAP HIGH RESOLUTION

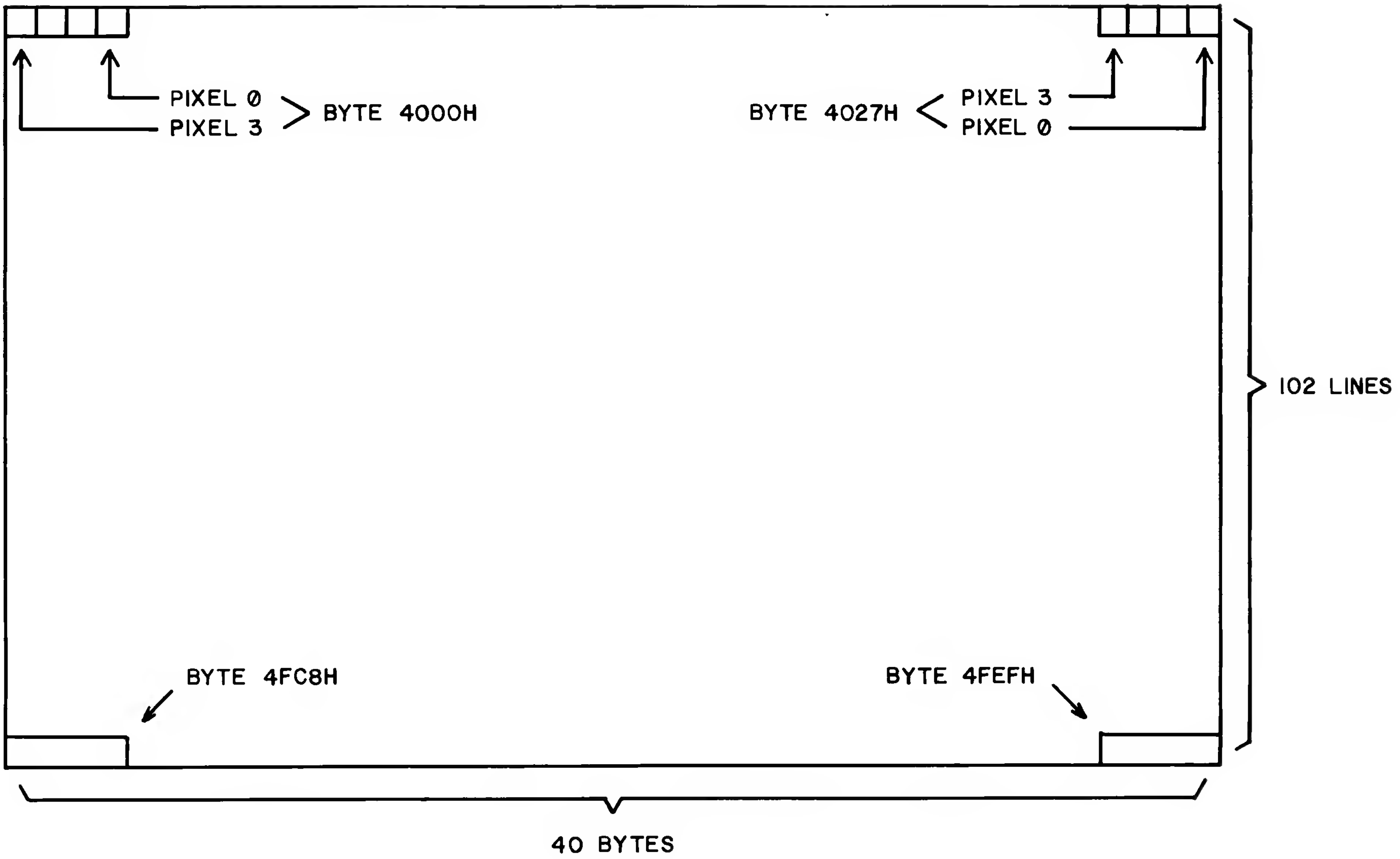
SCREEN MAP

In the Bally Professional Arcade, two bits of RAM are used to define a pixel on the screen. One 8-bit byte of RAM therefor defines four pixels on the screen.

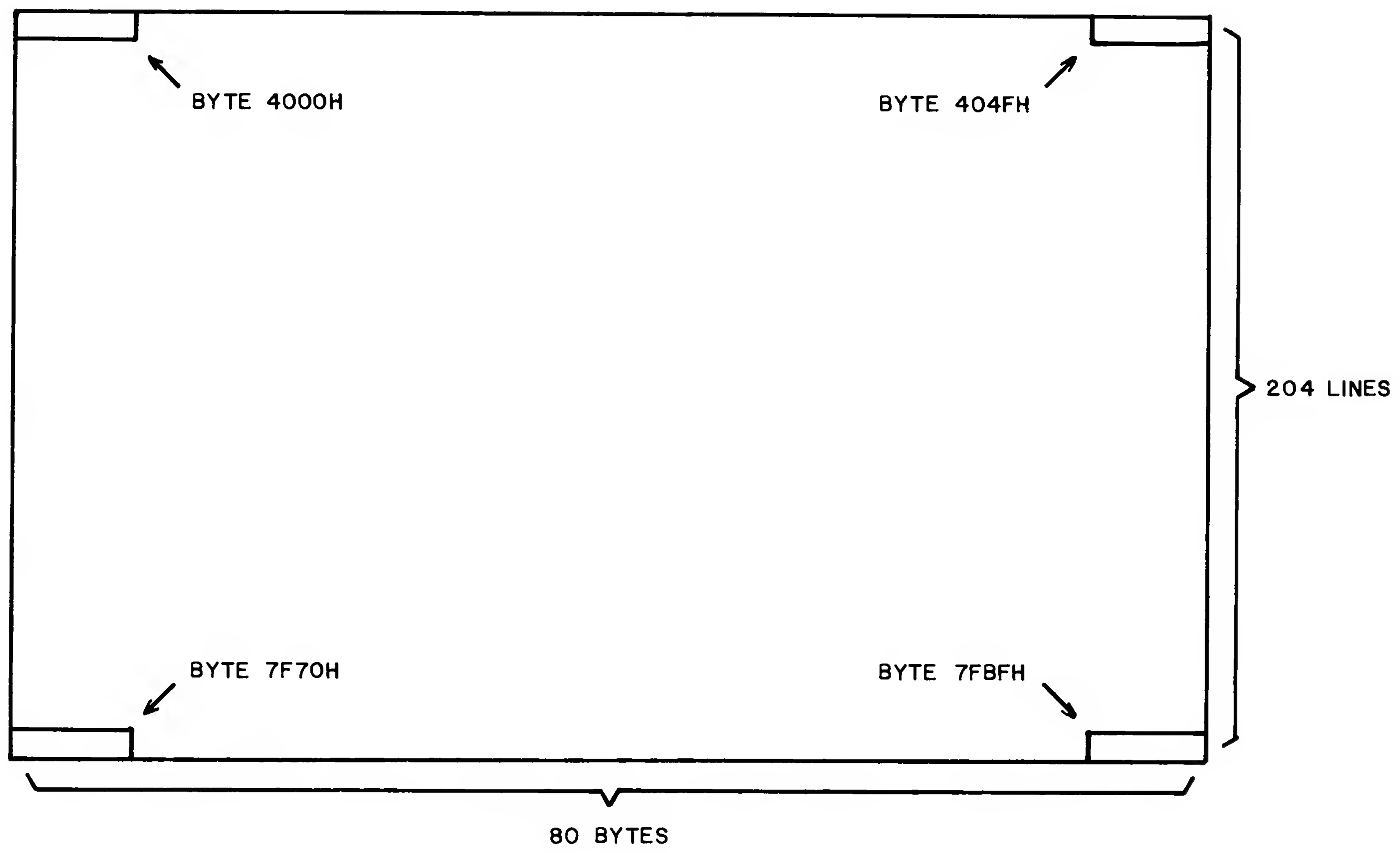
In the low-resolution model there are 40 bytes used to define a line of data. This gives a horizontal resolution of 160 pixels. The vertical resolution is 102 lines. The screen therefor requires 102 X 40 = 4,080 bytes. The remaining 16 bytes of the 4K RAM are used for scratch pad. More of the RAM can be used for scratchpad by blanking the screen before the 102nd line. This will be described later.

In the high-resolution model there are 80 bytes and 320 pixels per line. The 204 lines require 16,320 bytes of RAM. 64 bytes of the 16K RAM are left for scratch pad.

In both models the first byte of RAM is in the upper left-hand corner of the screen. As the RAM address increases, the position on the screen moves in the same directions as the TV scan; from left-to-right and from top-to-bottom. The four pixels in each byte are displayed with the least significant pixel, the one defined by bits Ø and 1, on the right.



SCREEN MAP LOW RESOLUTION



SCREEN MAP HIGH RESOLUTION

COLOR MAPPING

Two bits are used to represent each pixel on the screen. These two bits, along with the LEFT/RIGHT bit which is set by crossing the horizontal color boundary, map each pixel to one of eight different color registers. The value in the color register then defines the color and intensity of the pixel on the screen. The intensity of the pixel is defined by the three least significant bits of the register, ØØØ for darkest and 111 for lightest. The color is defined by the five most significant bits. The color registers are at output ports Ø through 7; register Ø at port Ø, register 1 at port 1, etc.

The color registers can be accessed as individual ports ar all eight can be accessed by one OTIR instruction. The OTIR instruction is to port BH (register C=BH) and register B should be set to 8. The eight bytes of data pointed to by HL will go to the color registers

HL → Memory Location	X	Color Register 7
	X+1	Color Register 6
	X+2	Color Register 5
	X+3	Color Register 4
	X+4	Color Register 3
	X+5	Color Register 2
	X+6	Color Register 1
	X+7	Color Register Ø

The horizontal color boundary (bits Ø-5 of port 9) defines the horizontal position of an imaginary vertical line on the screen. The boundary line can be position between any two adjacent bytes in the low-resolution system. The line is immediately to the left of the byte whose number is sent to bits Ø-5 of port 9. For example, if the horizontal color boundary is set to Ø, the line will be just to the left of byte Ø; if it is set to 20, the line will be between bytes 19 and 20 in the center of the screen.

If a pixel is to the left of the boundary, its LEFT/RIGHT bit is set to 1. The LEFT/RIGHT bit is set to Ø for pixels to the right of the boundary. Color registers Ø-3 are used for pixels to the right of the boundary and registers 4-7 are used for pixels to the left of the boundary.

In the high-resolution system, the boundary is placed in the same position on the screen but between different bytes. If the value X is sent to the horizontal color boundary, then the boundary will be between bytes 2X and 2X-1. If the value 20 is sent, the boundary will be between 39 and 40, in the center of the screen.

To put the entire screen, including the right side background, on the left side of the boundary, set the horizontal color boundary to 44.

BACKGROUND COLOR

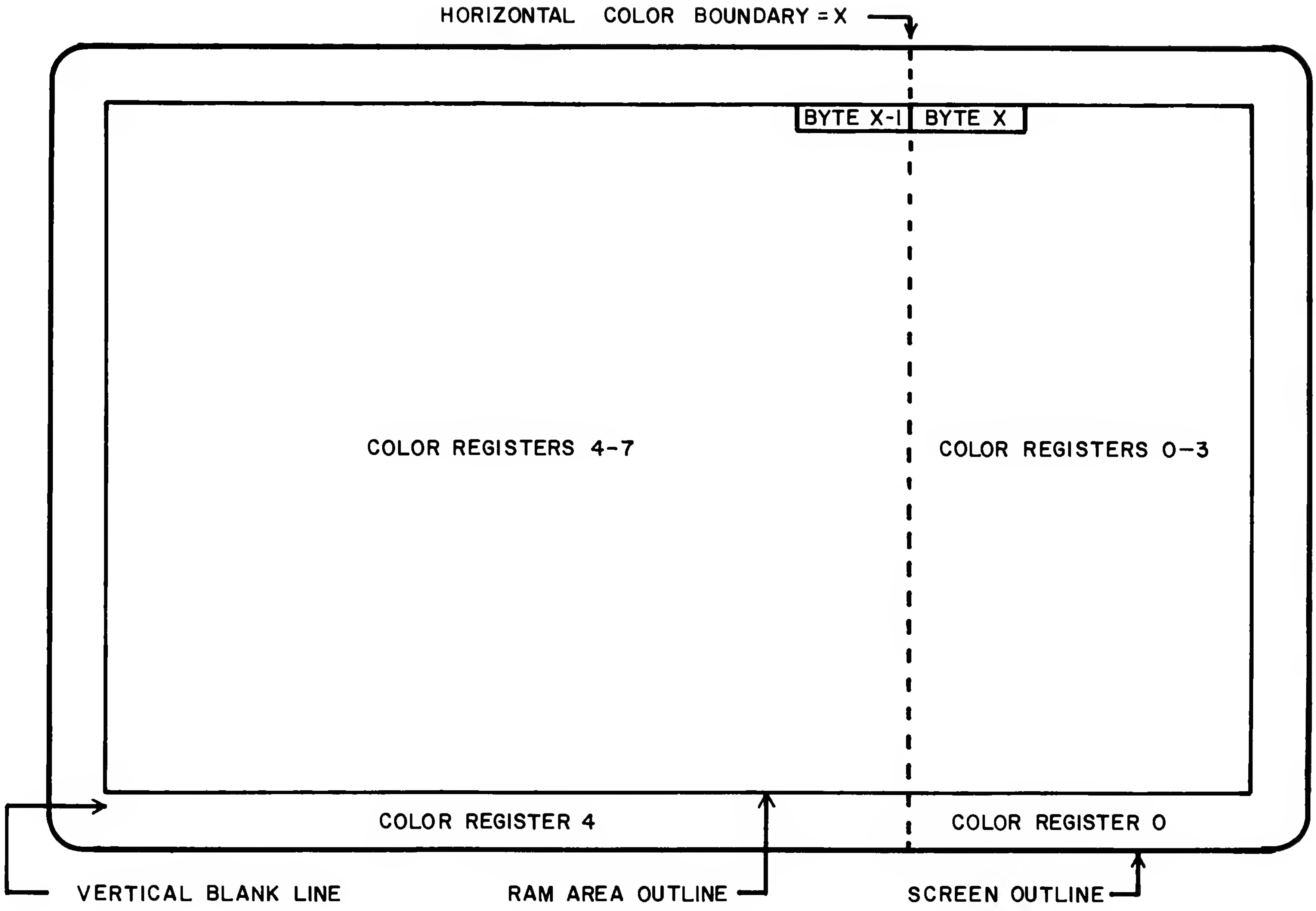
On most television the area defined by RAM is slightly smaller than the screen. There is generally extra space on all four sides of the RAM area. The color and intensity of this area is defined by the background color number (bits 6 and 7 of port 9). These two bits, along with the LEFT/RIGHT bit point to one of the color registers which determines the color and intensity.

VERTICAL BLANK

The Vertical Blank Register (output port AH) contains the line number on which vertical blanking will begin. In the low-resolution system bit Ø should be set to Ø and the line number should be in bits 1-7. In the high-resolution system the line number is in bits Ø-7. The background color will be displayed from the vertical blank line to the bottom of the screen. This allows the RAM that would normally be displayed in that area to be used for scratch pad. If the vertical blank register is set to Ø the entire RAM can be used for scratch pad. In a low-resolution system the register must be set to 101 or less; in a high-resolution system it must be set to 203 or less.

SUMMARY

The following color register map shows which color registers are used to define colors in different areas of the screen. The map assumes the background color is set to Ø. If it were set to 1 then color registers 1 and 5 would be used for background instead of Ø and 4. In the low-resolution system the color boundary is between bytes X and X-1. In the high-resolution system the boundary is between bytes 2X and 2X-1.



COLOR REGISTER MAP

INTERRUPT FEEDBACK

When the Z-80 acknowledges an interrupt it reads 8 bits of data from the data bus. It then uses this data as an instruction or an address. In the Bally Professional Arcade this data is determined by the contents of the interrupt feedback register (output port DH). In responding to a screen interrupt the contents of the interrupt feedback register are placed directly on the data bus. In responding to a light pen interrupt the lower four bits of the data bus are set to Ø and the upper four bits are the same as the corresponding bits of the feedback register.

INTERRUPT CONTROL BITS

In order for the Z-80 to be interrupted the internal interrupt enable flip-flop must be set by an EI instruction and one or two of the external interrupt enable bits must be set (output port EH). If bit 1 is set, light pen interrupts can occur. If bit 3 is set, screen interrupts can occur. If both bits are set, both interrupts can occur and the screen interrupt has higher priority.

The interrupt mode bits determine what happens if an interrupt occurs when the Z-80's interrupt enable flip-flop is not set. Each of the two interrupts may have a different mode. In mode Ø the Z-80 will continue to be interrupted until it finally enables interrupts and acknowledges the interrupt. In mode 1 the interrupt will be discarded if it is not acknowledged by the next instruction after it occured. If mode 1 is used the software must be designed such that the system will not be executing certain Z-80 instructions when the interrupt occurs. The opcodes of these instructions begin with CBH, DDH, EDH, and FDH.

The mode bit for light pen interrupt is bit Ø of port EH and the mode bit for screen interrupt is bit 2 of port EH.

SCREEN INTERRUPT

The purpose of the screen interrupt is to synchronize the software with the video system. The software must send a line number to the interrupt line register (output port FH). In the low-resolution system bit Ø is set to Ø and the line number is sent to bits 1-7. In the high-resolution system the line number is sent to bits 0-7. If the screen interrupt enable bit is set, the Z-80 will be interrupted when the video system completes scanning the line in the interrupt register. This interrupt can be used for timing since each line is scanned 60 times a second. It can also be used in conjunction with the color registers to make as many as 256 color-intensity combinations appear on the screen at the same time.

LIGHT PEN INTERRUPT

The light pen interrupt occurs when the light pen trigger is pressed and the video scan crosses the point on the screen where the light pen is. The interrupt routine can read two registers to determine the position of the light pen. The line number is read from the vertical feedback register (input port EH). In the high-resolution system the line number is in bits Ø-7. In the low-resolution system the line number is in bits 1-7, bit Ø should be ignored. The horizontal position of the light pen can be determined by reading input port FH and subtracting 8. In the low-resolution system the resultant value is the pixel number, Ø to 159. In the high-resolution system the resultant must be multiplied by two to give the pixel number, Ø to 358.

MAGIC REGISTER

As described earlier, the Magic System is enabled when data is written to a memory location (X) from Ø to 16K. A modified form of the data is actually written in memory location X+16K. The magic register (output port CH) determines how the data is modified. The purpose of each bit of the magic register is shown below.

Bit Ø	LSB of shift amount
1	MSB of shift amount
2	Rotate
3	Expand
4	OR
5	XOR
6	Flop

The order is which magic functions are performed is as follows: Expansion is done first; rotating or shifting; flopping; OR or XOR. As many as four can be used at any one time and any function can be bypassed. Rotate and shift as well as OR and XOR cannot be done at the same time.

EXPAND

The expander is used to expand the 8 bit data bus into 8 pixels (or 16 bits). It expands a Ø on the data bus into a two-bit pixel and a 1 into another two-bit pixel. Thus, two-color patterns can be stored in ROM in half the normal memory space.

During each memory write instruction using the expander, either the upper half or the lower half of the data bus is expanded. The half used is determined by the expand flip-flop. The flip-flop is reset by an output to the magic register and is toggled after each magic memory write. The upper half of the data bus is expanded when the flip-flop is Ø, and the lower half when the flip-flop is 1.

The expand register (output port 19H) determines the pixel values into which the data bus will be expanded. A Ø on the data bus will be expanded into the pixel defined by bits Ø and 1 of the expand register. A 1 on the data bus will be expanded into the pixel defined by bits 2 and 3 of the expand register.

The pixels generated by bit Ø or 4 of the data bus will be the least significant pixel of the expanded byte. The most significant pixel will come from bit 3 or 7.

SHIFTER

The shifter, flopper, and rotator operate on pixels rather than bits. Each byte is thought of as containing four pixels, each of which has one of four values. The four pixels are referred to as PØ, P1, P2, and P3. PØ is composed of the first two bits of the byte.

The shifter shifts data Ø, 1, 2, or 3 pixels to the right. The shift amount is determined by bits Ø and 1 of the magic register. The pixels that are shifted out of one byte are shifted into the next byte. Ø's are shifted into the first byte of a sequence. The shifter assumes the first byte of a sequence is the first magic memory write after an output to the magic register. Each sequence must be initialized by an output to the magic register and data cannot be sent to the magic register in the middle of a sequence.

FLOPPER

The output of the flopper is a mirror image of it's input. Pixel Ø and 3 exchange values as do pixel 1 and 2.

The diagrams on the following page show examples of shifting and flopping.

BYTE 0				BYTE 1				BYTE 2				
P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	P9	P8	ORIGINAL DATA
0	P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	P9	SHIFT 1
0	0	P3	P2	P1	P0	P7	P6	P5	P4	P11	P10	SHIFT 2
0	0	0	P3	P2	P1	P0	P7	P6	P5	P4	P11	SHIFT 3
P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	FLOPPED

SHIFTER — FLOPPER

ROTATOR

The rotator is used to rotate a 4 X 4 pixel image 90° in a clockwise direction. The rotator is initialized by an output to the magic register and will re-initialize itself after every eight writes to magic memory. To perform a rotation, the following procedure must be performed twice. Write the top byte of the unrotated image to a location in magic memory. Write the next byte to the first location plus 80, the next byte to the first location plus 160, and the last byte to the first location plus 240. After eight writes the data will appear in RAM and on the screen rotated 90° from the original image.

The rotator can only be used in commercial mode.

The diagram on the following page shows an example of rotating.

P3	P2	P1	P0
P7	P6	P5	P4
P11	P10	P9	P8
P15	P14	P13	P12

ORIGINAL

P15	P11	P7	P3
P14	P10	P6	P2
P13	P9	P5	P1
P12	P8	P4	P0

ROTATED

ROTATOR

OR AND XOR

These functions operate on a byte as 8-bits rather than four pixels. When the OR function is used in writing data to RAM, the input to the OR circuit is ORed with the contents of the RAM location being accessed. The resultant is then written in RAM.

The XOR function operates in the same way except that the data is XORed instead of ORed.

INTERCEPT

Software reads the intercept register (input port 8H) to determine if an intercept occured on an OR or XOR write. An intercept is defined as the writing of a non-zero pixel in a pixel location that previously contained a non-zero pixel. A non-zero pixel is a pixel with a value of Ø1, 1Ø, or 11. A 1 in the intercept register means an intercept has occured. Bits Ø - 3 give the intercept information for all OR or XOR writes since the last input from the intercept register. An input from the intercept register resets these bits. A bit is set to 1 if an intercept occurs in the appropriate position and will not be reset until after the next intercept register input.

Bit	
Ø	Intercept in pixel 3 in an OR or XOR write since last reset
1	Intercept in pixel 2 in an OR or XOR write since last reset
2	Intercept in pixel 1 in an OR or XOR write since last reset
3	Intercept in pixel Ø in an OR or XOR write since last reset
4	Intercept in pixel 3 in last OR or XOR write
5	Intercept in pixel 2 in last OR or XOR write
6	Intercept in pixel 1 in last OR or XOR write
7	Intercept in pixel Ø in last OR or XOR write

PLAYER INPUT

The system will accomodate up to four player control handles at once. Each handle has five switches and a potentiometer. The switches are read by the Z-80 on input ports 10H - 13H and are not debounced. The switches are normally open and normally feedback Ø's.

The signals from the potentiometers are changed to digital information by an 8-bit Analog-to-Digital Convertor. The four pots are on input ports 1CH - 1FH. All Ø's are fedback when the pot is turned fully counter-clockwise and all 1's when turned fully clockwise.

The 24-button keypad is read on bits Ø-5 of ports 14H-17H. The data is normally Ø and if more than one button is depressed, the data should be ignored. The keypad will not send back the proper data if any of the player control switches are closed. Here again, the buttons are not debounced.

Player control inputs are shown on the following page.

PLAYER INPUT

PORT ↓	7	6	5	4	3	2	1	0	← BIT
10H				TRIG	RIGHT	LEFT	DOWN	UP	PLAYER 1
11H				TRIG	RIGHT	LEFT	DOWN	UP	PLAYER 2
12H				TRIG	RIGHT	LEFT	DOWN	UP	PLAYER 3
13H				TRIG	RIGHT	LEFT	DOWN	UP	PLAYER 4
14H			=	+	−	X	÷	%	KEYPAD
15H			.	3	6	9	CH	∨	KEYPAD
16H			0	2	5	8	MS	∧	KEYPAD
17H			CE	1	4	7	MR	C	KEYPAD
1CH	← POT →								PLAYER 1
1DH	← POT →								PLAYER 2
1EH	← POT →								PLAYER 3
1FH	← POT →								PLAYER 4

MASTER OSCILLATOR

The frequency of the master oscillator is determined by the contents of several output ports. Port 1ØH sets the master frequency. It is given by the following formula:

$$F_m = \frac{1789}{\text{PORT } 10H + 1} \text{ Khz}$$

If bit 4 of output port 15H is set to 1, the master oscillator frequency will be modulated by noise. The amount of modulation will be set by the 8-bit noise volume register, output port 17H.

If bit 4 of output port 15H is set to Ø, the frequency of the master oscillator will be modulated by a constant value to give a vibrato effect. The amount of modulation will be set by the vibrato depth register (the first 6 bits of output port 14H). The speed of modulation is set by the vibrato speed register (upper 2 bits of output port 14H); ØØ for fastest and 11 for slowest.

Frequency modulation is accomplished by adding a modulation value to the contents of port 1ØH and sending the result to the master oscillator frequency generator. In noise modulation, the modulation value is an 8-bit word from the noise generator. If a bit in the noise volume register is set to Ø, the corresponding bit in the modulation value word will be set to Ø. In vibrato modulation, the modulation value alternates between Ø and the contents of the vibrato volume register.

Modulation can be completely disabled by setting the master volume to Ø if noise modulation is being used, or by setting the vibrato depth to Ø when vibrato is used.

TONES

The system contains three tone generators each clocked by the same master oscillator. The frequency of Tone A is set by output port 11H, Tone B by output port 12H, and Tone C by output port 13H. The frequency is given by the following formula:

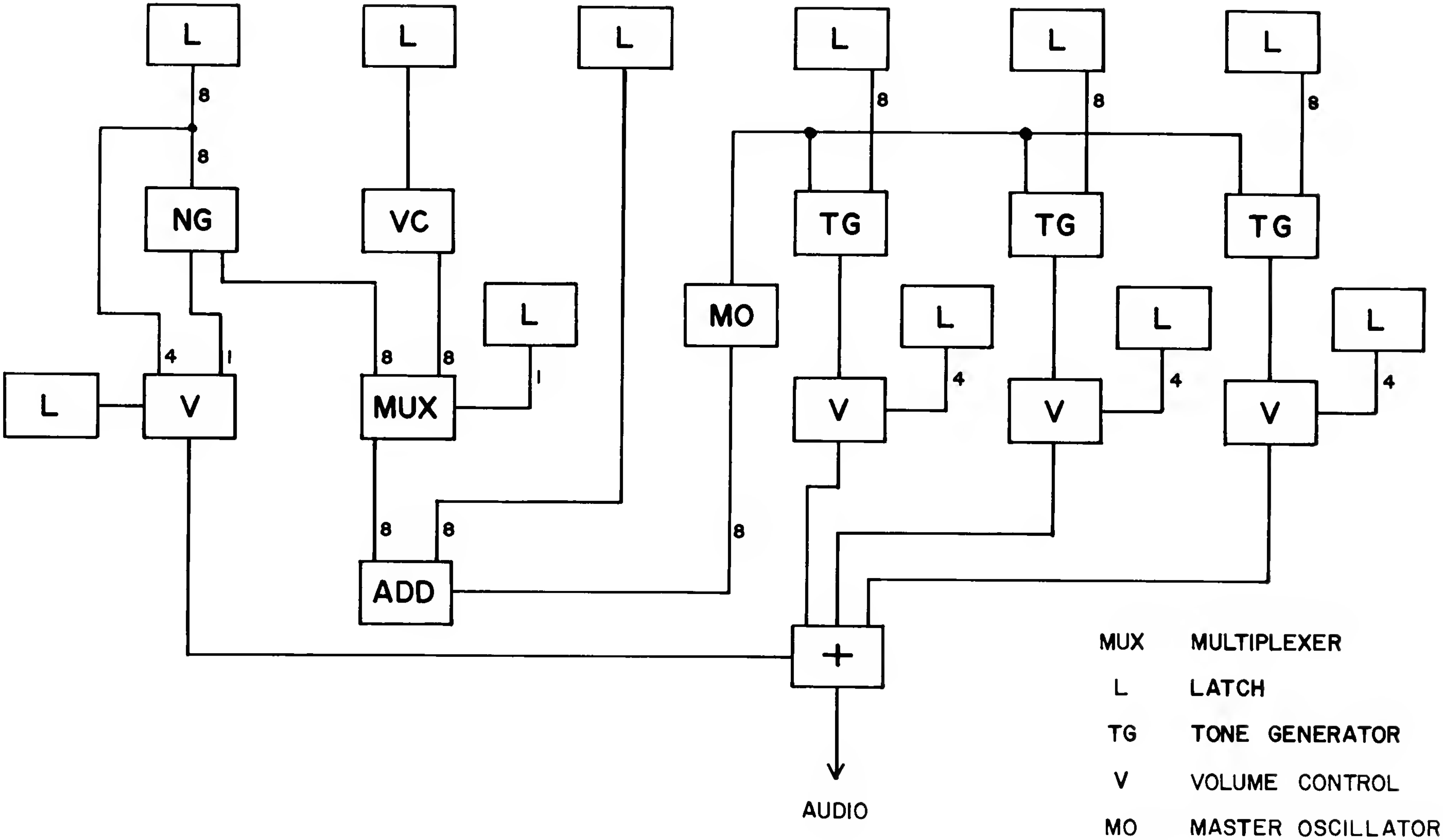
$$F_t = \frac{F_m}{2(\text{contents of TONE PORT}+1)} = \frac{894}{(\text{PORT } 10\text{H}+1)(\text{contents of TONE PORT}+1)} \text{ Khz}$$

The tone volumes are controlled by output ports 15H and 16H. The lower 4 bits of port 16H set Tone A Volume, the upper 4 bits sets Tone B Volume. The lower 4 bits of port 15H sets Tone C Volume. Noise can be mixed with the tones by setting bit 5 of port 15H to 1. In this case the noise volume is given by the upper 4 bits of port 17H. In all cases a volume of Ø is silence and a volume of all 1's is loudest.

SOUND BLOCK TRANSFER

All 8 bytes of sound control can be sent to the audio circuit with one OTIR instruction. Register C should be sent to 18H, register B to 8H and HL pointing to the 8 bytes of data. The data pointed to by HL goes to port 17H and the next 7 bytes of data goes to ports 16H through 1ØH.

HL →	Memory Location	X	Data-to-port	17H
		X+1	Data-to-port	16H
		X+2	Data-to-port	15H
		X+3	Data-to-port	14H
		X+4	Data-to-port	13H
		X+5	Data-to-port	12H
		X+6	Data-to-port	11H
		X+7	Data-to-port	1ØH



MUX	MULTIPLEXER
L	LATCH
TG	TONE GENERATOR
V	VOLUME CONTROL
MO	MASTER OSCILLATOR
VC	VIBRATO CONTROL
NG	NOISE GENERATOR
ADD	ADDER

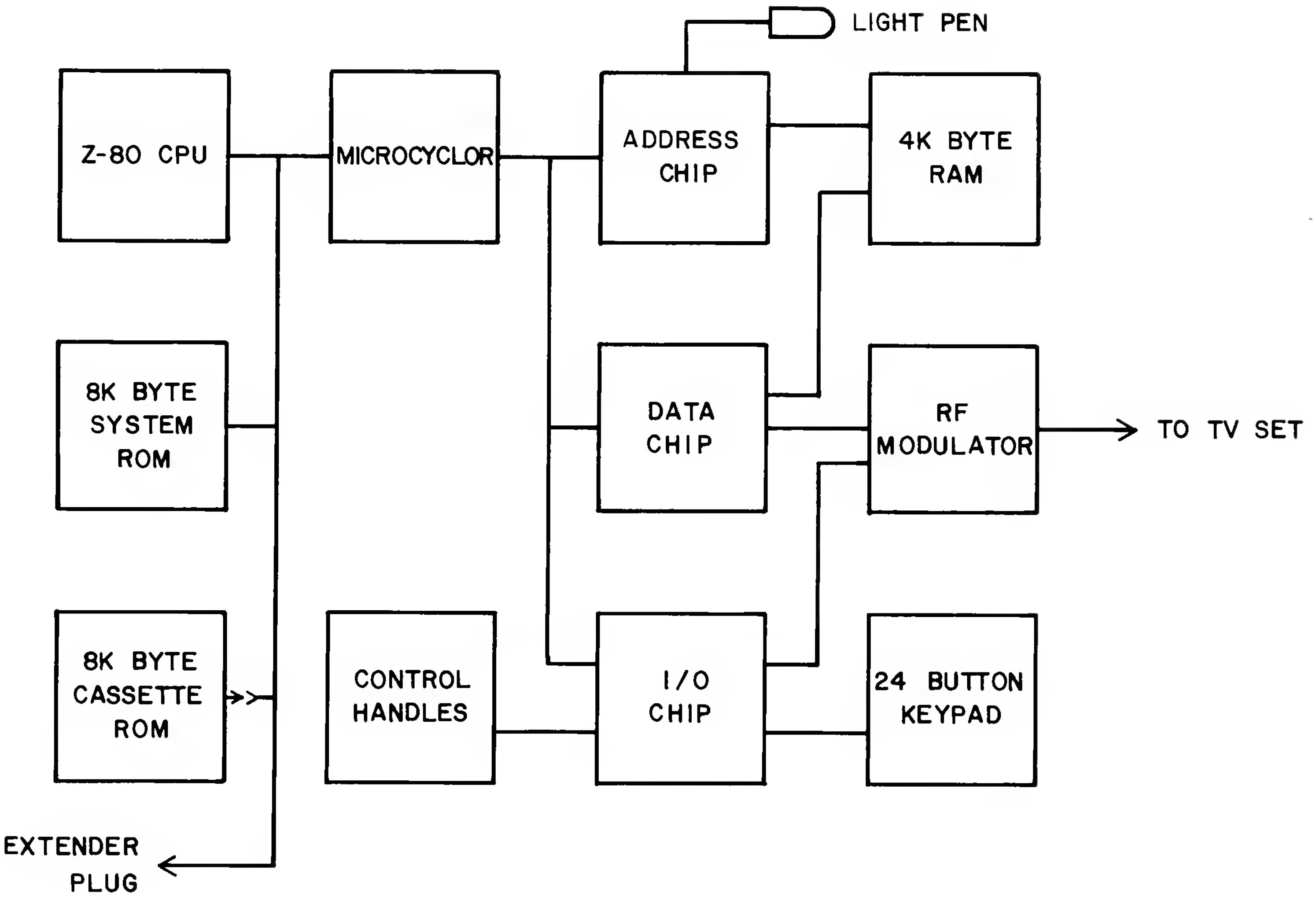
AUDIO GENERATOR BLOCK DIAGRAM

OUTPUT PORTS

PORT NUMBER	FUNCTION
ØH	Color Register Ø
1H	Color Register 1
2H	Color Register 2
3H	Color Register 3
4H	Color Register 4
5H	Color Register 5
6H	Color Register 6
7H	Color Register 7
8H	Low/High Resolution
9H	Horizontal Color Boundary, Background Color
AH	Vertical Blank Register
BH	Color Block Transfer
CH	Magic Register
DH	Interrupt Feedback Register
EH	Interrupt Enable and Mode
FH	Interrupt Line
1ØH	Master Oscillator
11H	Tone A Frequency
12H	Tone B Frequency
13H	Tone C Frequency
14H	Vibrato Register
15H	Tone C Volume, Noise Modulation Control
16H	Tone A Volume, Tone B Volume
17H	Noise Volume Register
18H	Sound Block Transfer
19H	Expand Register

INPUT PORTS

PORT NUMBER	FUNCTION
8H	Intercept Feedback
EH	Vertical Line Feedback
FH	Horizontal Address Feedback
1ØH	Player 1 Handle
11H	Player 2 Handle
12H	Player 3 Handle
13H	Player 4 Handle
14H	Keypad Column Ø (right)
15H	Keypad Column 1
16H	Keypad Column 2
17H	Keypad Column 3 (left)



SYSTEM BLOCK DIAGRAM

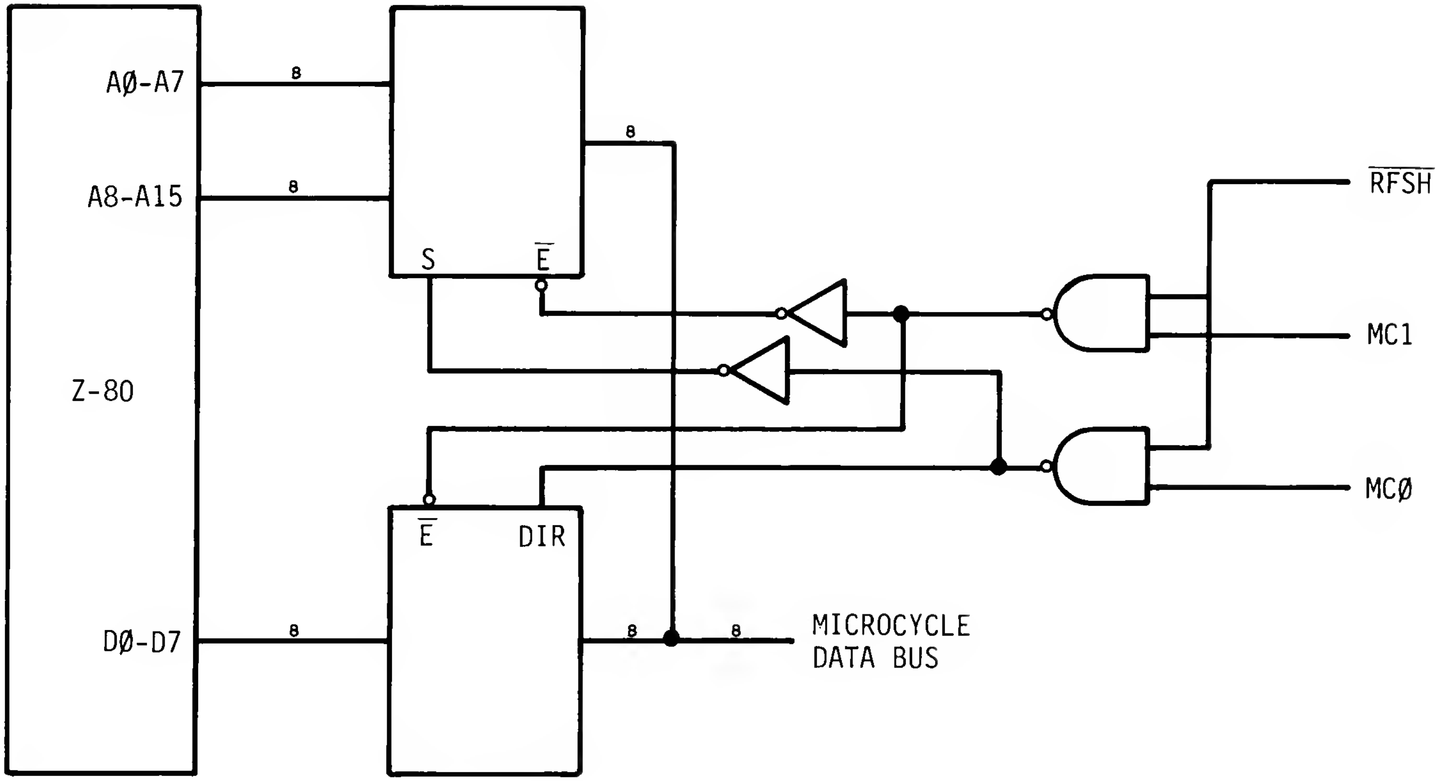
MICROCYCLER

The purpose of the microcycler is to combine the 16-bit Address Bus and the 8-bit Data Bus from the Z-8Ø into one 8-bit Microcycle Data Bus to the Data Chip, Address Chip, and I/O Chip. This was done to reduce the pin count on the custom chips.

The Microcycle Data Bus can be in any of four modes. Its mode is controlled by MCØ and MC1 coming from the Data Chip and $\overline{\text{RFSH}}$ from the Z-8Ø. The modes are shown below.

$\overline{\text{RFSH}}$	MCØ	MC1	Microcycle Data Bus Contents
Ø	Ø	Ø	AØ - A7 from Z-8Ø
Ø	Ø	1	AØ - A7 from Z-8Ø
Ø	1	Ø	AØ - A7 from Z-8Ø
Ø	1	1	AØ - A7 from Z-8Ø
1	Ø	Ø	AØ - A7 from Z-8Ø
1	Ø	1	A8 - A15 from Z-8Ø
1	1	Ø	DØ - D7 from Z-8Ø
1	1	1	DØ - D7 to Z-8Ø

MCØ and MC1 change 140 nsec after the rising edge of Φ. Their changes are shown in the timing diagrams of various instruction cycles.



MICROCYCLER BLOCK DIAGRAM

ADDRESS CHIP DESCRIPTION

The Microcycle Decoder generates twelve bits of Z-8Ø address from the 8-bit Microcycle Data Bus. This address is then fed through MUX I and MUX II to MAØ-5 which go to the RAM. The Scan Address Generator generates a 12-bit address which is used to read video data from the RAM. This address goes from Ø to FFFH once every frame (1/60 sec.).

MUX I sends either the Scan Address or Z-8Ø Address to its 12 outputs. An output of the Scan Address Generator controls MUX I. If the Scan Address Generator and the Z-8Ø request a memory cycle at the same time, the Scan Address Generator will have higher priority and the Z-8Ø will be required to wait (by the $\overline{\text{WAIT}}$ output). The Scan Address Generator never requires the memory for more than one consecutive memory cycle, so the Z-8Ø is never required to wait for the memory for more than one cycle. HORIZ DR and VERT DR synchronize the Scan Address Generator with the Data Chip and the TV Scan.

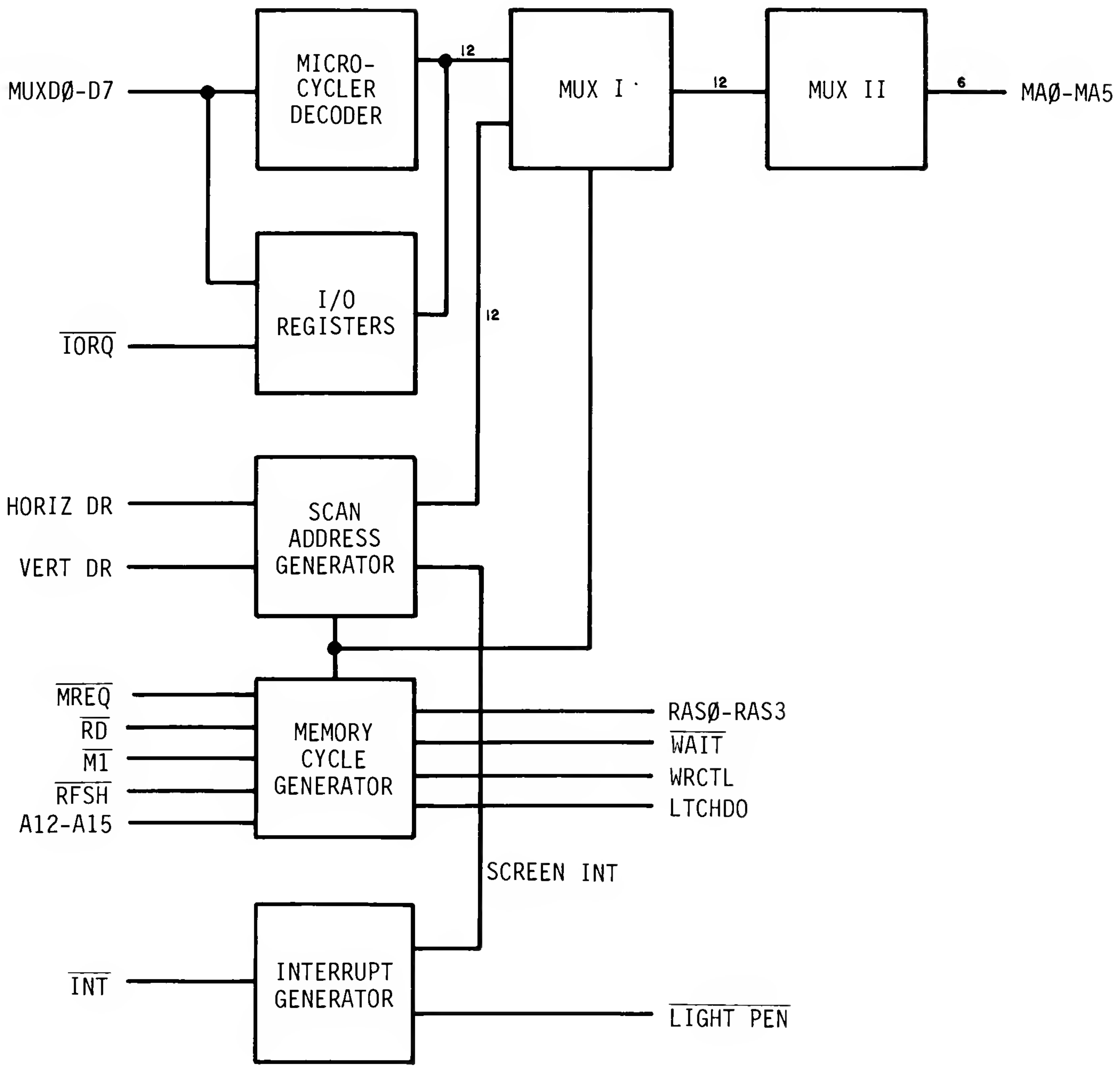
The purpose of MUX II is to multiplex its 12 inputs to the six address bits in the two time slices required for 4K x 1 16 pin RAMS.

The Memory Cycle Generator controls memory cycles generated by either the Z-8Ø or Scan Address Generator. $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, $\overline{\text{MI}}$, $\overline{\text{RFSH}}$, and A12-A15 are from the Z-8Ø. A12-A15 are fed directly from the Z-8Ø because if they were brought out of the microcycle decoder, they would arrive too late in the memory cycle. The RASØ - RAS3 outputs are used to activate memory cycles. In the consumer game, only RASØ is used to one bank of RAM (4K x 8). In the commercial game,all four RAS's are used to control four banks of RAM (16K x 8). WRCTL and LTCHDO are control signals to the Data Chip. WRCTL tells the Data Chip when to place data to be written to memory on the memory data bus. LTCHDO tells the Data Chip when valid data from RAM is present on the memory data bus.

As mentioned earlier, $\overline{\text{WAIT}}$ is generated when the Z-8Ø and Scan Address Generator both request memory at the same time. $\overline{\text{WAIT}}$ is also generated for one cycle every time the Z-8Ø requests a memory access, even if there is no conflict with the Scan Address. This is because the microcycler slows down Z-8Ø memory accesses. The Z-8Ø address bus and data bus must time share the microcycle bus so the Z-8Ø data reaches the microcycle bus very late in the memory cycle.

The INT Generator generates two types of interrupts to the Z-8Ø; Light Pen and Screen interrupts. A screen interrupt is generated when screen interrupts are enabled and the TV scan completes a certain line on the screen (from Ø to 255). The line at which the interrupt will occur is determined by the Z-8Ø. This interrupt can be used for timing since the TV rescans every line once every 1/60 sec. A light pen interrupt occurs when the light pen interrupt is enabled and $\overline{\text{LIGHT PEN}}$ goes low. The current scan address is saved in latches in the Scan Address Generator. The Z-8Ø can read the contents of these latches to determine the scan address at the time $\overline{\text{LIGHT PEN}}$ was activated and thus the position of the light pen on the screen.

The I/O Decode circuit is used during Z-8Ø input and output instructions. Z-8Ø input instructions are used to read the scan address after light pen interrupts. Output instructions are used to enable the two interrupts and set the line number for screen interrupts.



ADDRESS CHIP BLOCK DIAGRAM

DATA CHIP DESCRIPTION

The TV Sync Generator uses 7M and $\overline{\text{7M}}$ (7.159090 Mhz square waves) to generate NTSC standard sync and blank to be sent to the Video Generator. It also generates HORIZ DR and VERT DR for synchronization with the Address Chip. HORIZ DR occurs once every horizontal line (63.5 usec), and VERT DR occurs once every frame (16.6 msec).

The Shift Register loads parallel data from the memory data bus (MDØ - MD7) and shifts it out of its two serial outputs. The TV sync Generator controls when data is loaded or shifted. In a consumer game,the two outputs of the shift register are sent through MUX I to MUX II. In a commercial game, SERIAL Ø and SERIAL 1 are sent through the MUX I to MUX II. The two bits from MUX I select 8 bits to be sent through MUX II to the Video Generator. These 8 bits then determine the analog values of VIDEO, R-Y, and B-Y. 2.5V is a 2.5V D C reference level.

The Clock Generator generates ØG and $\overline{\text{PX}}$ from 7M. These are the clocks for the rest of the system. The frequency of $\overline{\text{PX}}$ is half that of 7M and the frequency of ØG is half that of $\overline{\text{PX}}$.

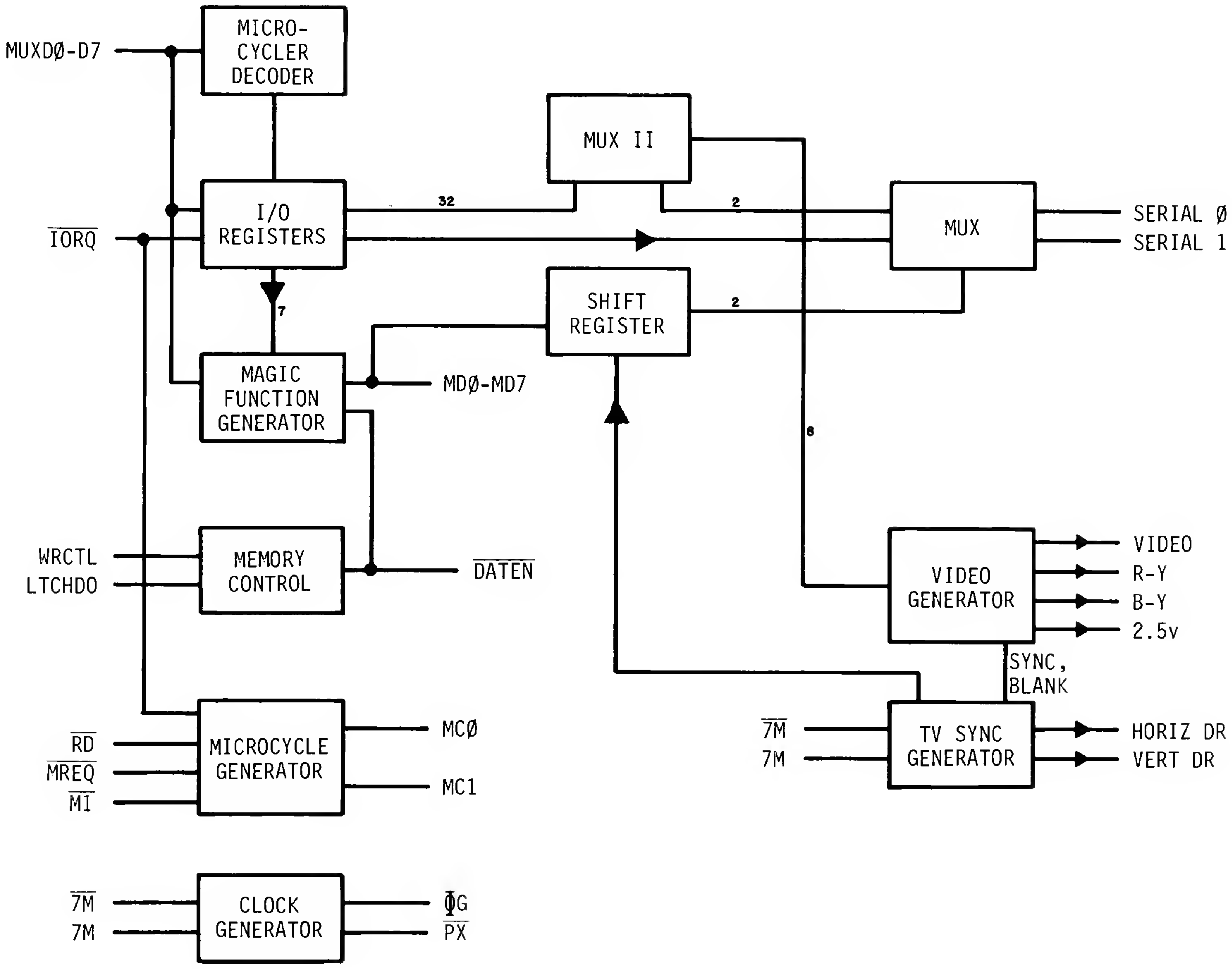
The Microcycle Generator generates the microcycle control bits, MCØ and MC1, from $\overline{\text{IORQ}}$, $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, and $\overline{\text{M1}}$, all from the Z-8Ø.

In memory write cycles WRCTL is activated and the Memory Control circuit generates $\overline{\text{DATEN}}$. The Magic Function Generator takes the data from the Z-8Ø on MUXDØ - D7 and transfers it to MDØ - MD7. If a Magic write is being done, the Magic Function Generator will modify the data as required before it places it on the memory data bus.

A Magic write is a memory write cycle in which data is written to a location, (X) from Ø to 16K. All memory from Ø to 16K is ROM and cannot be modified. The data is modified by the Magic Function Generator and is written to location X + 16K. The way in which the data is modified is determined by the 7 bits coming from the I/O registers.

In memory reads, data is transferred from MDØ - MD7 to MUXDØ - MUXD7. Also, LTCHDO is activated which causes the data from RAM to be latched up in a register in the Magic Function Generator. This latched data is used in some magic functions.

The I/O registers are loaded by output instructions from the Z-8Ø just as in the Address Chip.



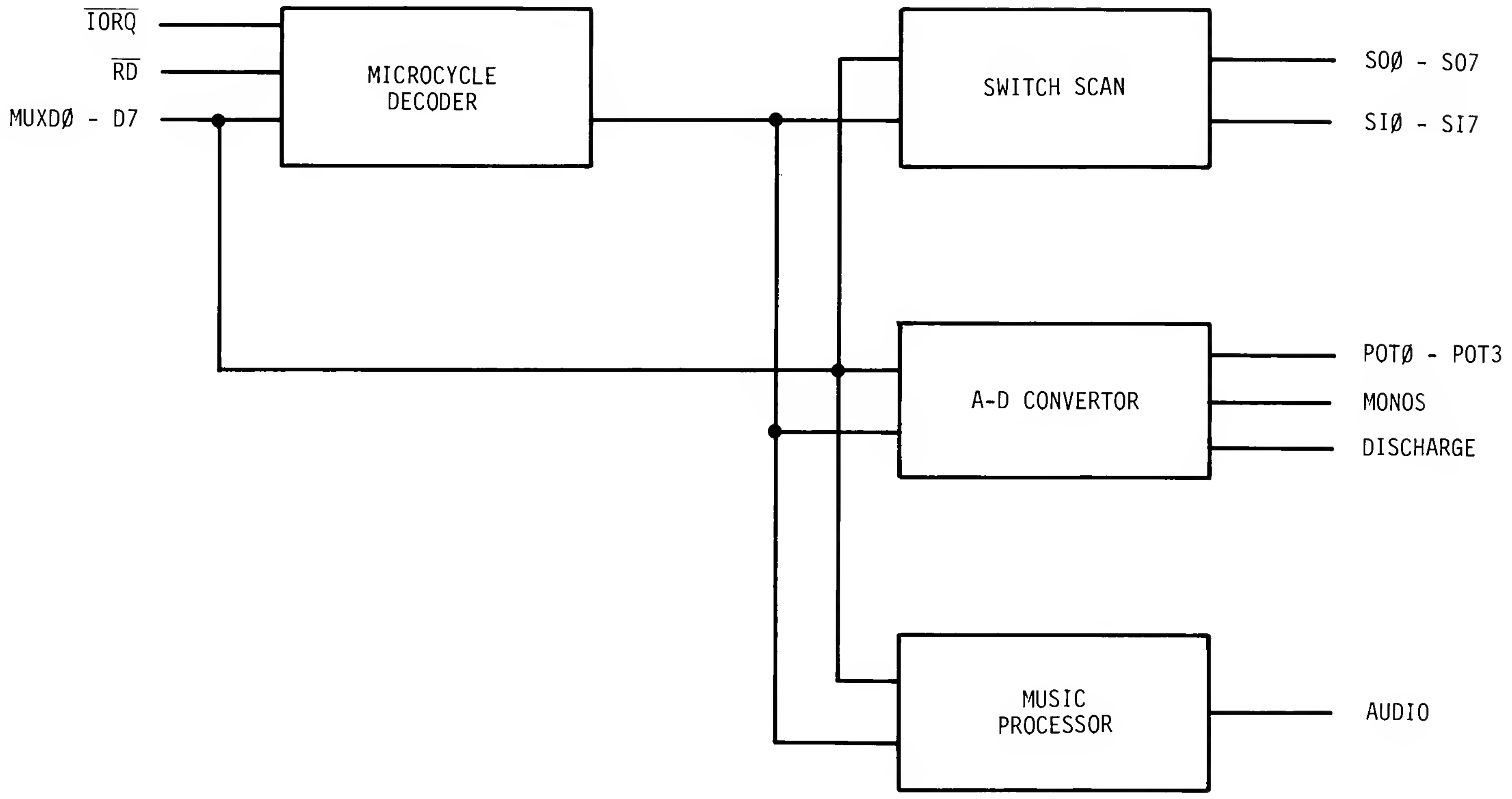
DATA CHIP BLOCK DIAGRAM

I/O CHIP DESCRIPTION

The Z-8Ø communicates with the I/O Chip through input and output instructions. The state of an 8 x 8 switch matrix can be read through the Switch Scan circuit. When an input instruction is executed, one of the SOØ-SO7 lines will be activated. When a line is activated, the switch matrix will feed back eight bits of data on SIØ-SI7. This data is in turn fed to the Z-8Ø through MUXDØ - MUXD7.

The Z-8Ø can read the position of four potentiometers (pots) through the A-D Converter circuit. The pots are continuously scanned by the A-D Converter and the results of the conversions are stored in a RAM in the A-D Converter circuit. The Z-8Ø simply reads this RAM with input instructions.

The Z-8Ø loads data into the Music Processor with output instructions. This data determines the characteristics of the audio that is generated. The Music Processor is described in detail below.



I/O CHIP BLOCK DIAGRAM

MUSIC PROCESSOR

The music processor can be divided into two sections. The first section generates the Master Oscillator Frequency and the second section uses the Master Oscillator Frequency to generate tone frequencies and the analog audio output. The contents of all registers in the Music Processor are set by output instructions from the Z-8Ø.

Master Oscillator Frequency is a square wave whose frequency is determined by the 8 binary inputs to the Master Oscillator. This 8-bit word is the sum of the contents of the Master Oscillator Register and the output of the MUX. The MUX is controlled by MUX REG.

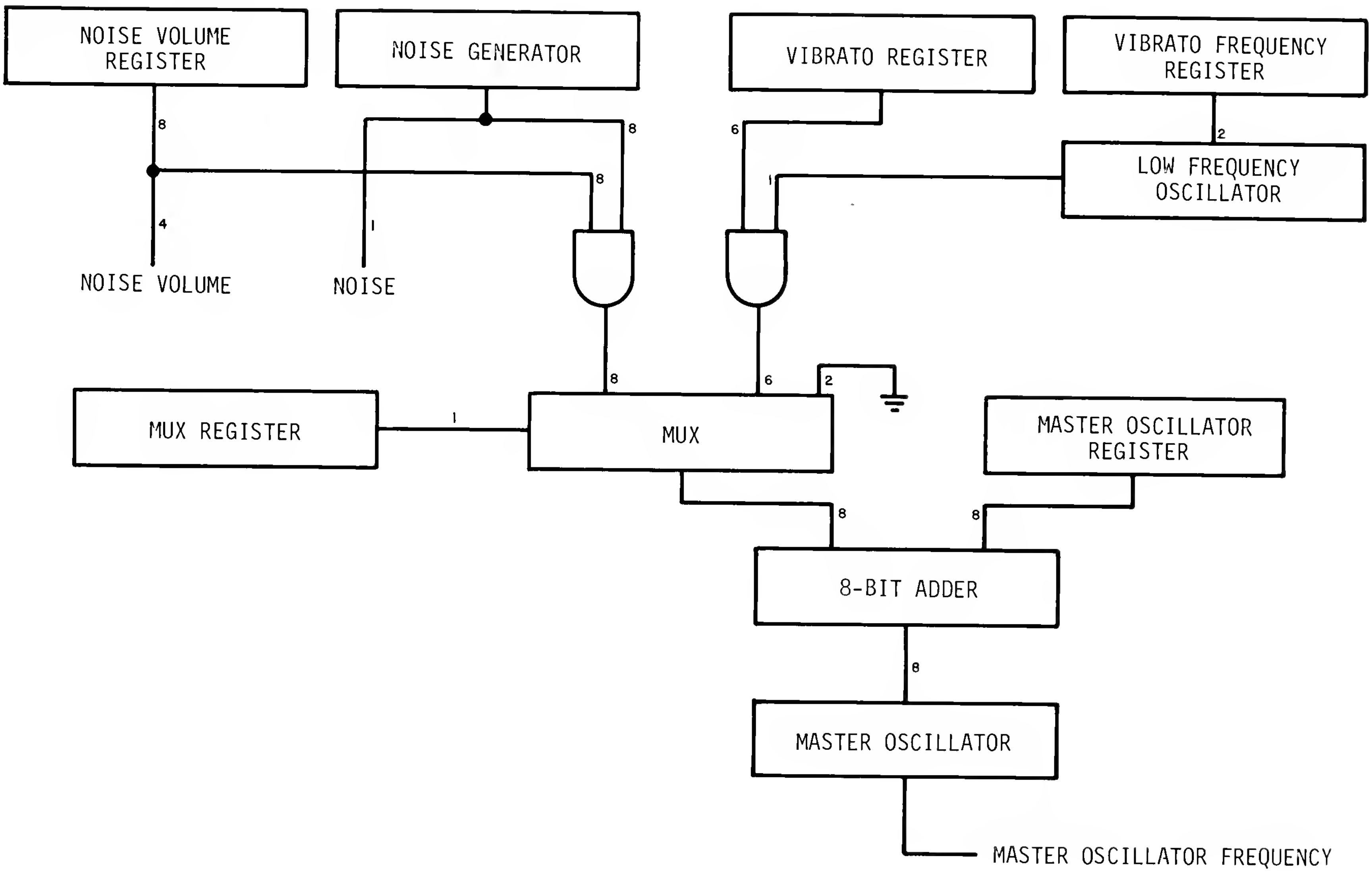
If MUX REG contains Ø,then data from the Vibrato System will be fed through the MUX. The two bits from the Vibrato Frequency Register determine the frequency of the square wave output of the Low Frequency Oscillator. The 6-bit word at the output of the AND gates oscillates between Ø and the contents of the Vibrato Register. The frequency of oscillation is determined by the contents of the Vibrato Frequency Register. The 6-bit word, along with two ground bits are fed through the MUX to the Adder. This causes the Master Oscillator Frequency to be modulated between two values thus giving a vibrato effect.

If MUX REG contains 1 ,then data from the Noise System will be fed through the MUX. The 8-bit word from the Noise Volume Register determines which bits from the Noise Generator will be present at the output of the AND gates.

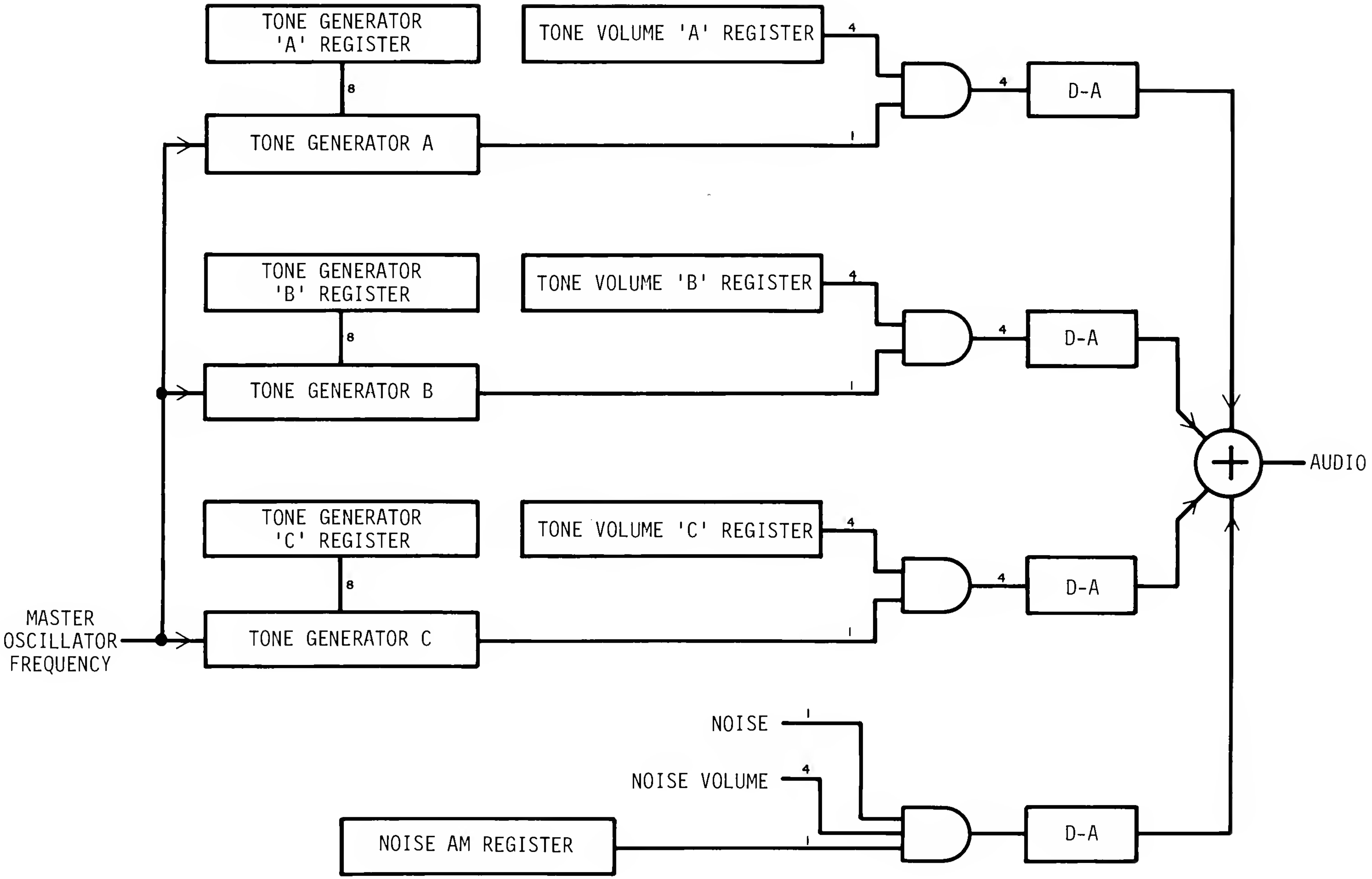
If a bit in the Noise Volume Register is ∅, then the corresponding bit at the output of the AND gates will be ∅. If a bit in the Noise Volume Register is 1, then the corresponding bit at the output of the AND gates will be noise from the Noise Generator. This 8-bit word is sent through the MUX to the Adder. The Master Oscillator Frequency is modulated by noise.

In the second part of the Music Processor, the square wave from the Master Oscillator is fed to three Tone Generator circuits which produce square waves at their outputs. The frequency of their outputs is determined by the contents of their Tone Generator Register and Master Oscillator Frequency. The 4-bit words at the output of the AND gates oscillate between ∅ and the contents of the Tone Volume Register. These 4-bit words are sent to D-A Converters whose outputs oscillate between GND and a positive analog voltage determined by the contents of the Tone Volume Register.

One Noise bit and four Noise Volume bits from the first section of the Music Processor are fed to a set of AND gates. This set of AND gates operates the same way as the AND gates for the tones, except that the Noise AM Register must contain a 1 for the outputs of the AND gates to oscillate. The analog outputs of the four D-A Converters are summed to produce the single audio output.



MASTER OSCILLATOR



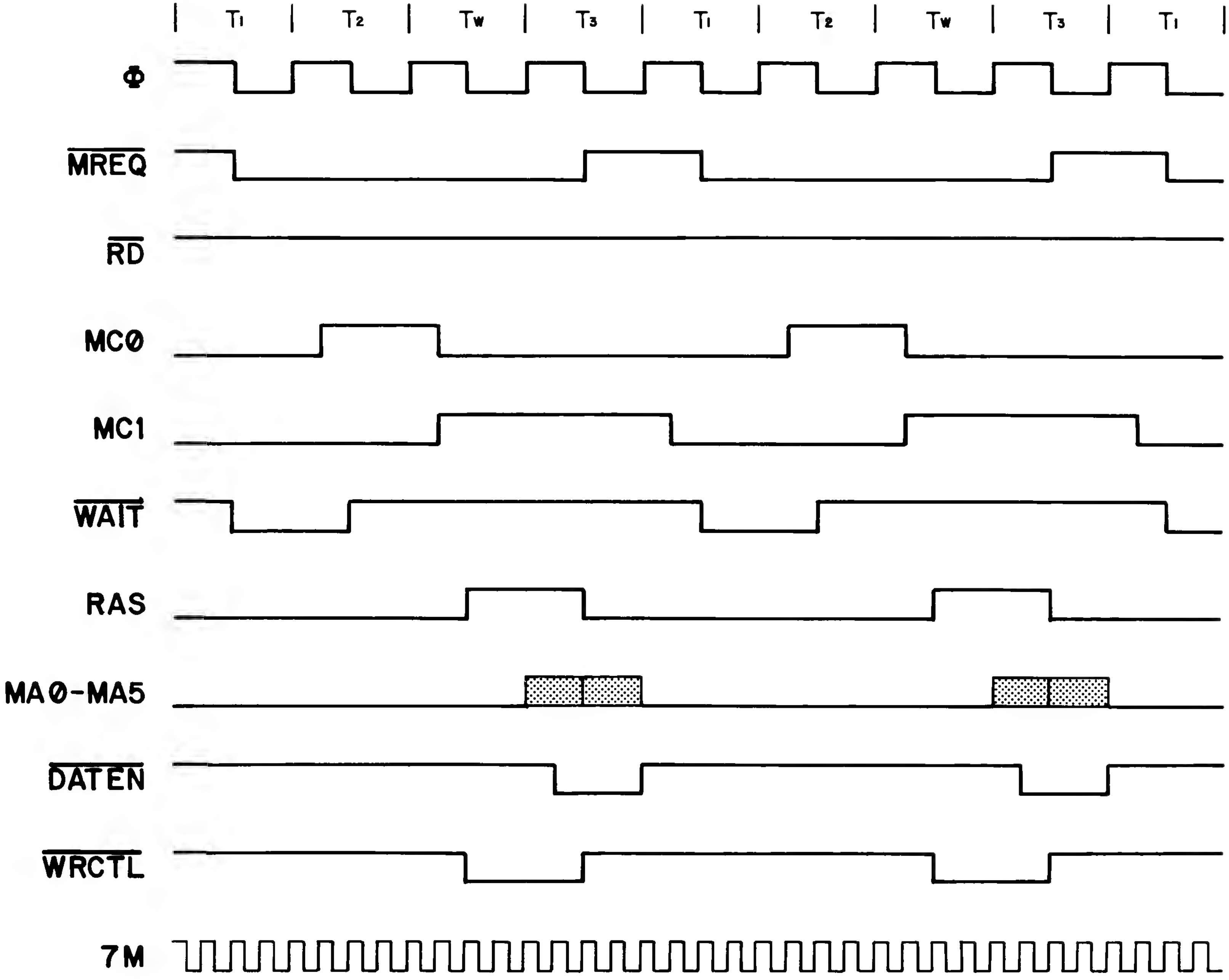
TONE GENERATORS

CUSTOM CHIP TIMING

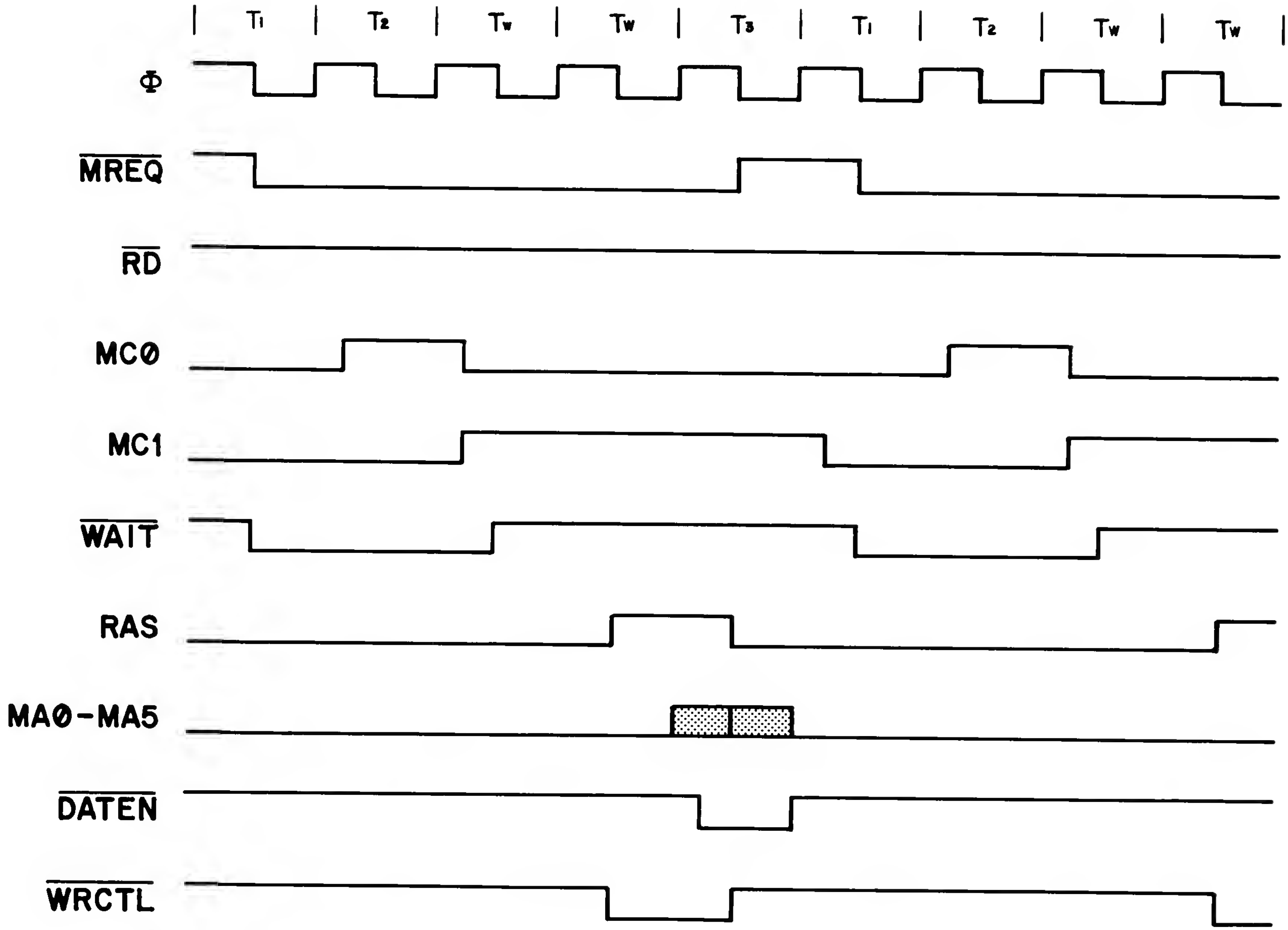
The following diagrams show the relationship of various signals in the system during different types of operations. Delays are shown to be zero nsec from the clock edge which causes the transition. The actual delay is given in "Electrical Specification for Midway Custom Circuits".

MUXDØ - MUXD7 is a 8-bit bidirectional address and data bus for the custom chips. By using this technique 16 bits of address and 8 bits of data can be sent to the custom chips on 8 wires. The state of the bus is determined by MCØ and MC1 from the data chip and $\overline{\text{RFSH}}$ from the Z-8Ø.

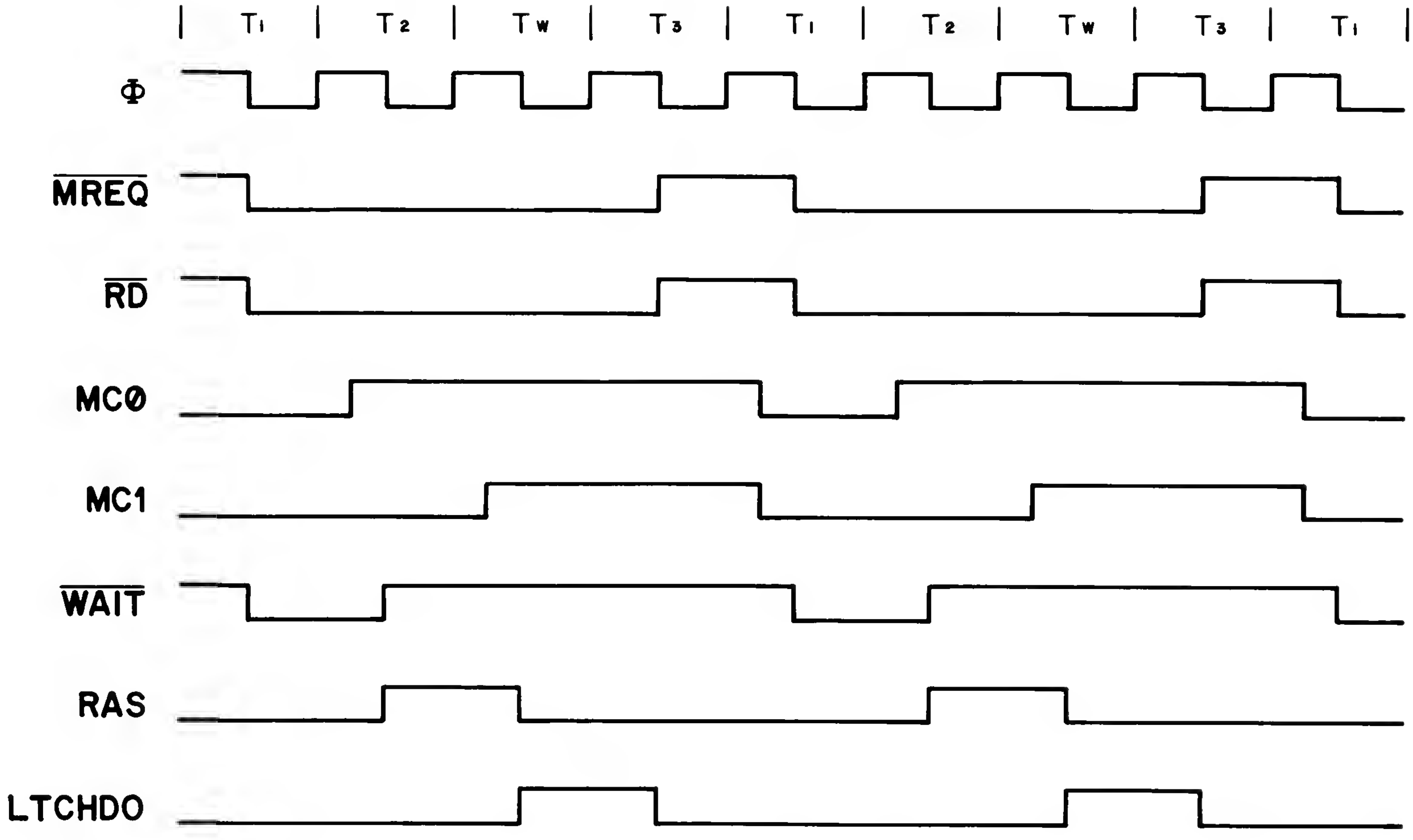
$\overline{\text{RFSH}}$	MC1	MCØ	
L	L	L	AØ - A7 to custom chips.
L	L	H	AØ - A7 to custom chips
L	H	L	AØ - A7 to custom chips
L	H	H	AØ - A7 to custom chips
H	L	L	AØ - A7 to custom chips
H	L	H	A8 - A15 to custom chips
H	H	L	DØ - D7 to custom chips
H	H	H	DØ - D7 from custom chips



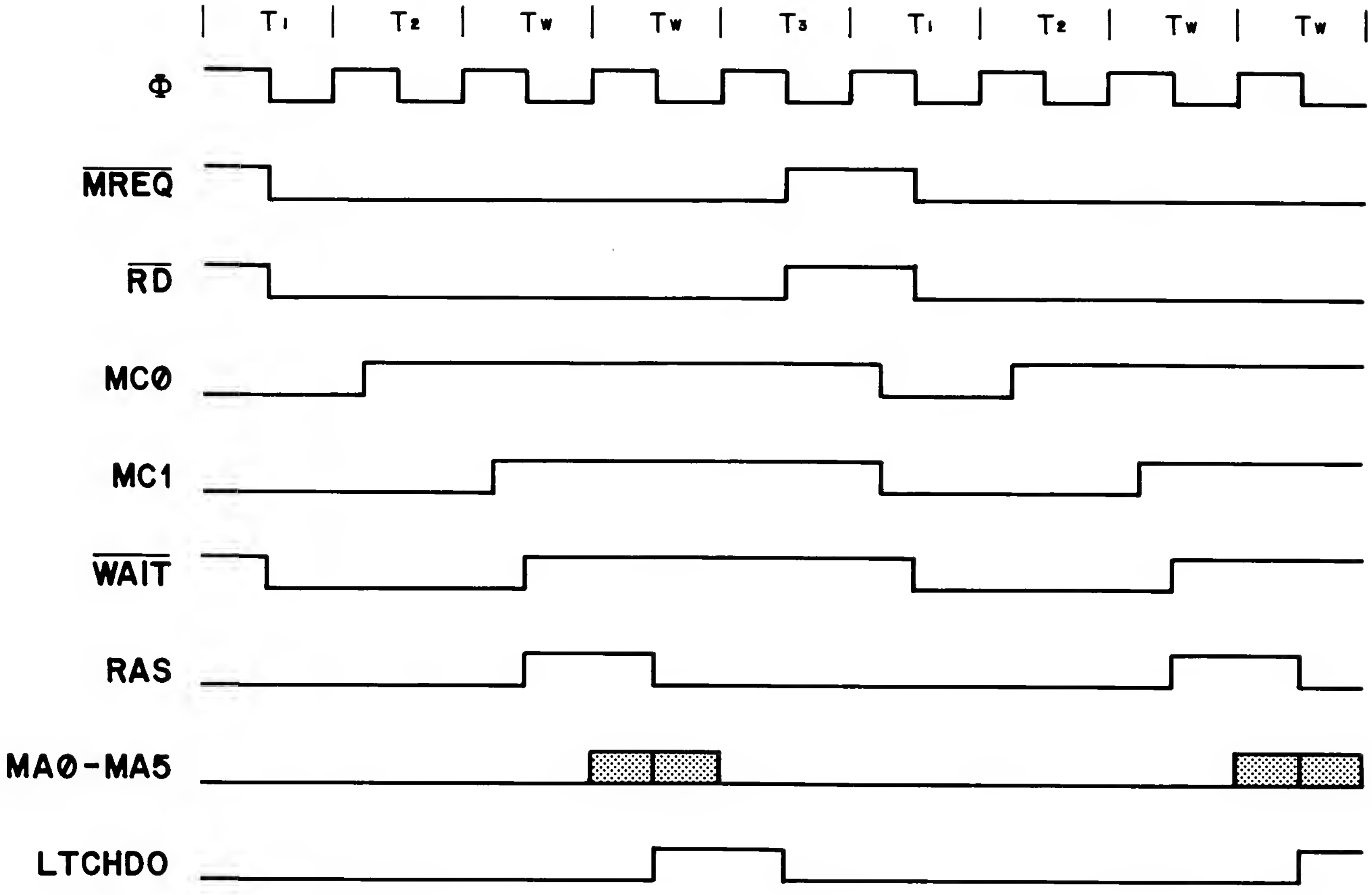
MEMORY WRITE WITHOUT EXTRA WAIT STATE



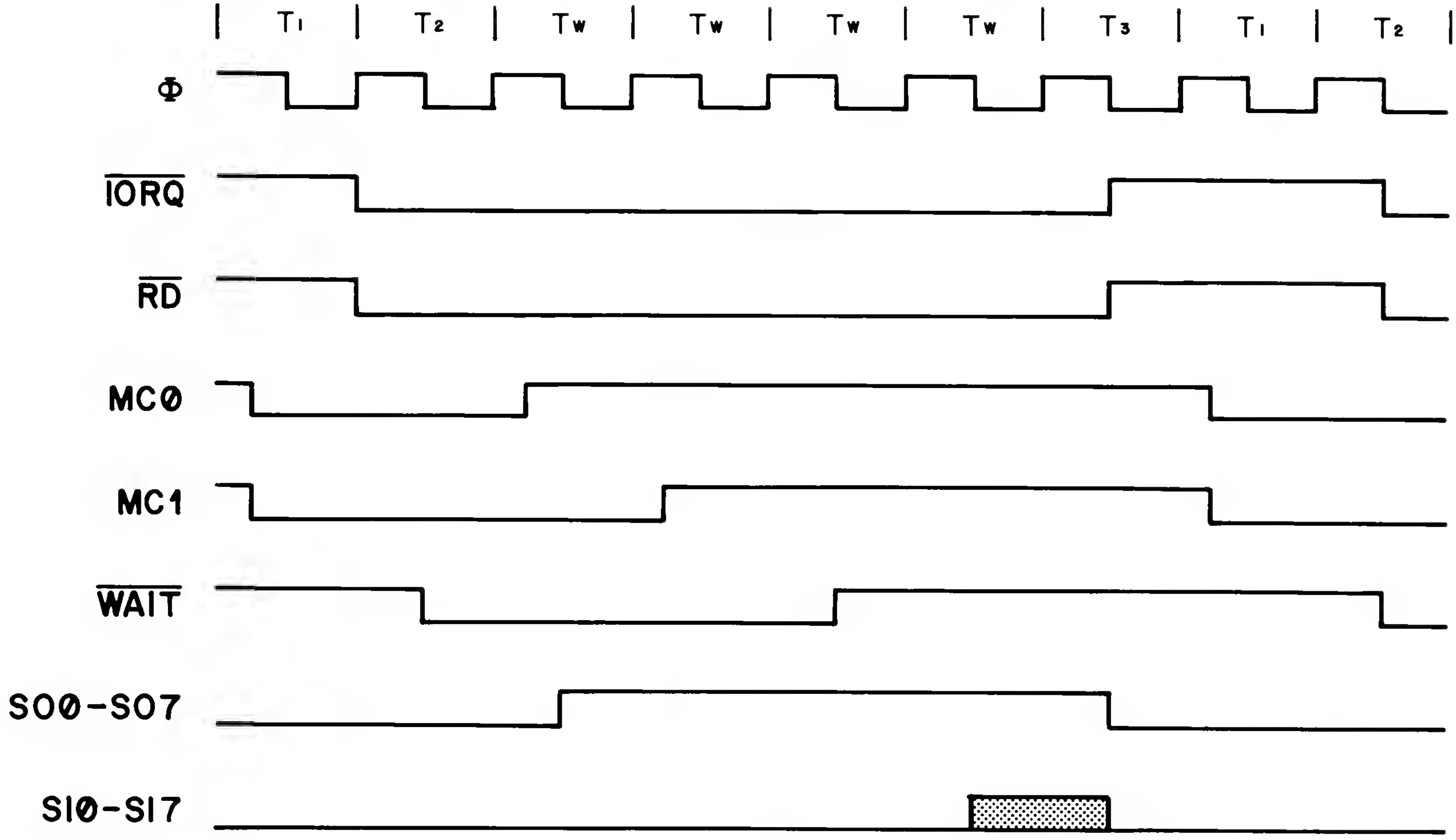
MEMORY WRITE WITH VIDEO WAIT STATE



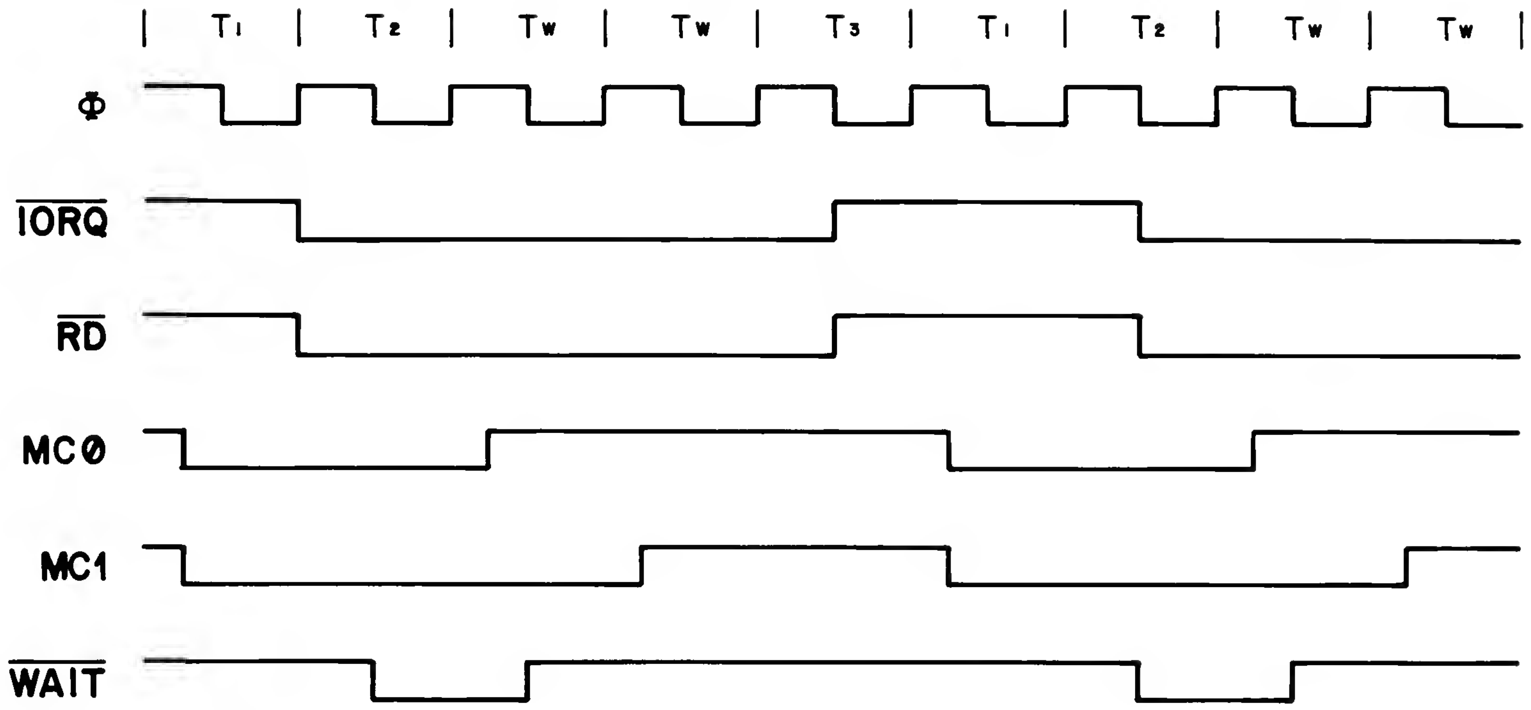
MEMORY READ WITHOUT EXTRA WAIT STATE



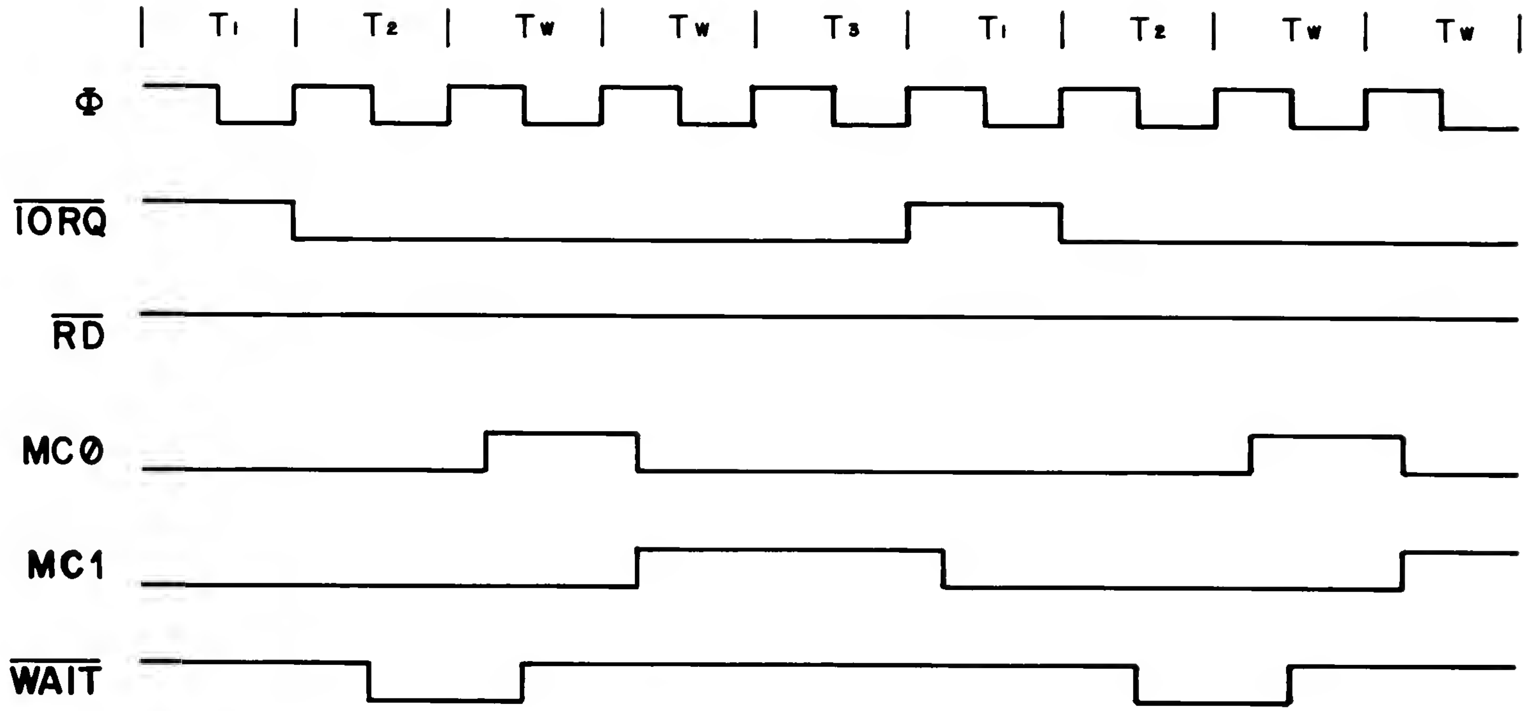
MEMORY READ WITH VIDEO WAIT STATE



I/O READ FROM PORT 1ØH-17H



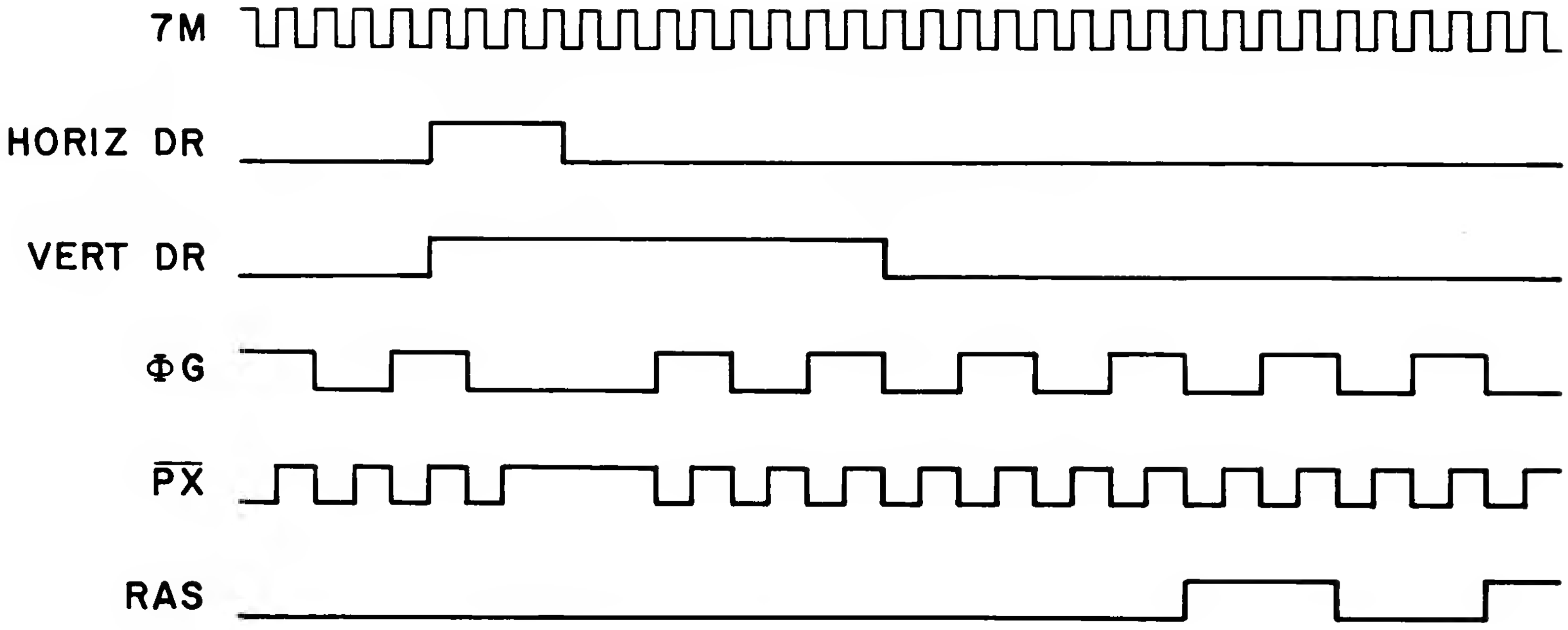
I/O READ FROM OTHER THAN PORT 1ØH-17H



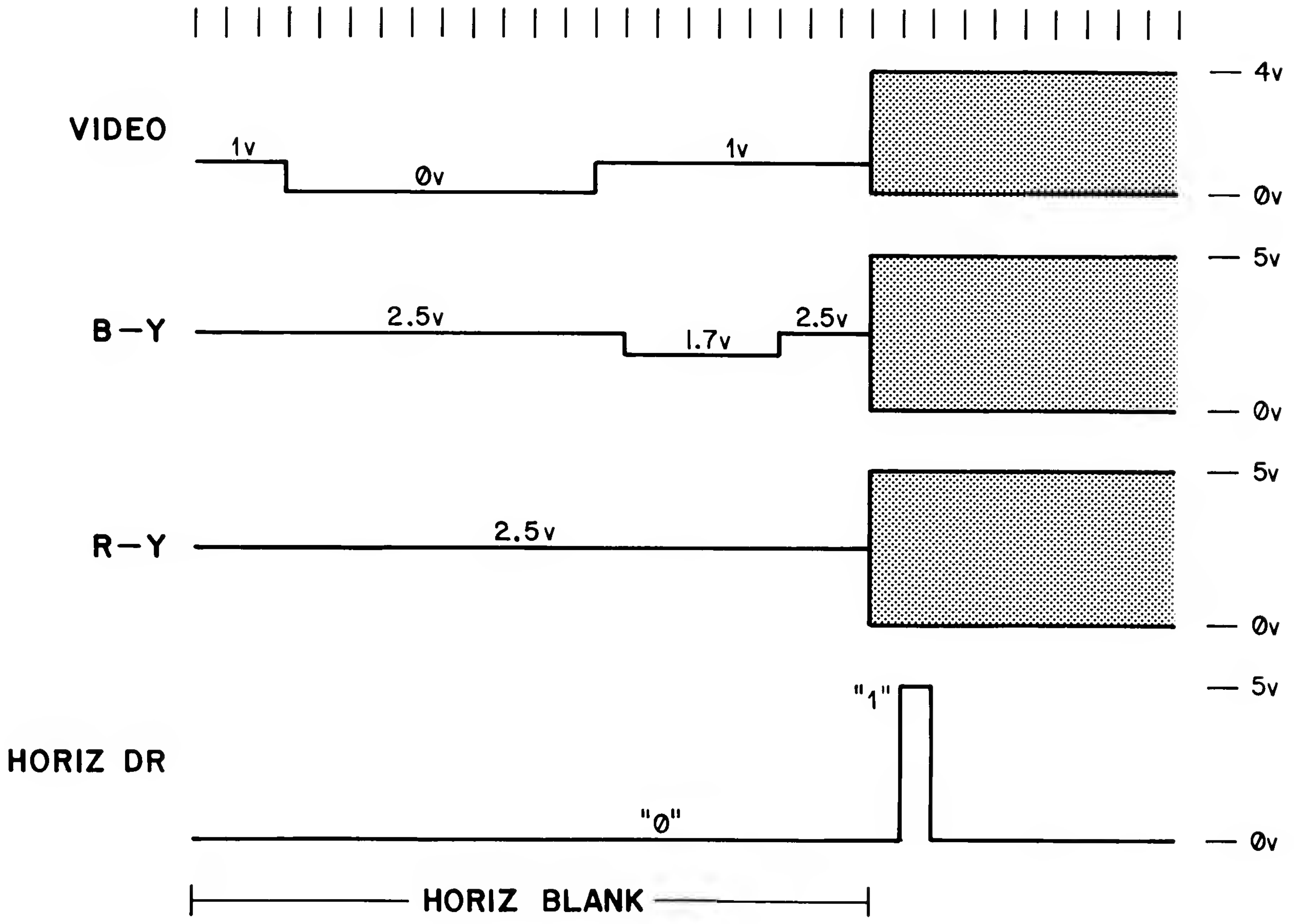
I/O WRITE

VIDEO TIMING

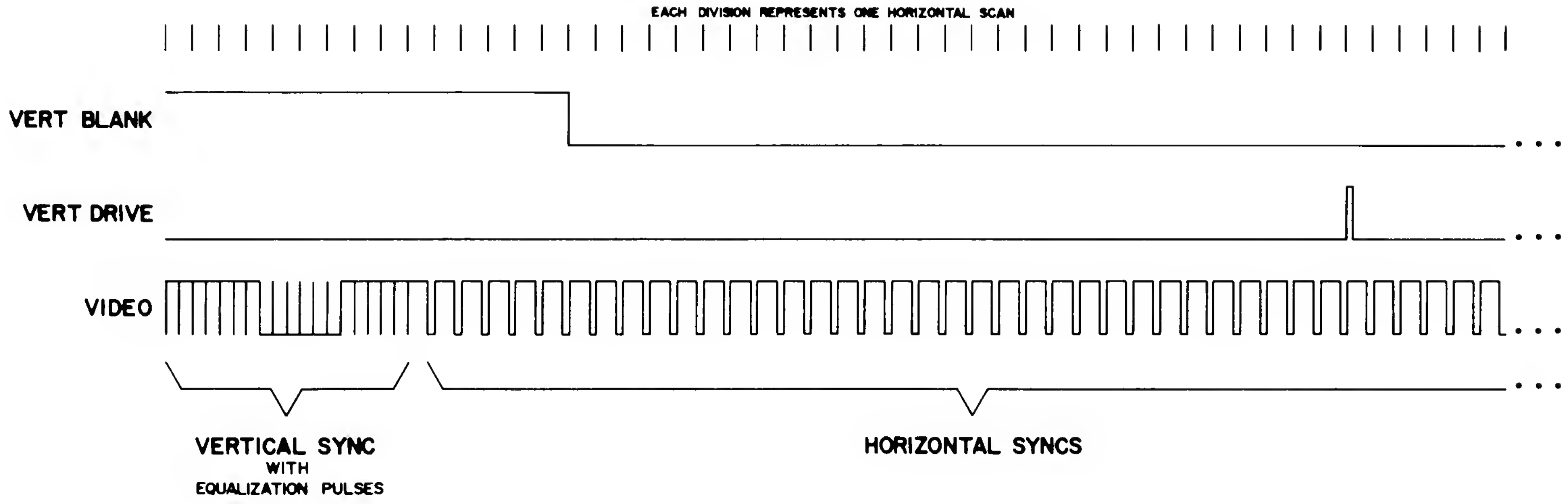
The frequency of $\overline{PX}$ is half that of 7M and the Ø is one-fourth 7M. There are 455 cycles of 7M per horizontal line and 113 3/4 Φ cycles per line. Because of the extra 3/4 cycle Ø must be resynchronized at the beginning of each line. This is done by stalling Ø for 3 cycles of 7M. $\overline{PX}$ is also stalled for the same amount of time. The timing relationship is shown below. The diagram also shows the relationship of VERT DR to HORIZ DR. The two RAS pulses shown are the first two video RAS signals of a line, each line contains forty.



RELATIONSHIP BETWEEN 7M, HORIZ DR, VERT DR, ΦG, $\overline{PX}$ AND RAS



RELATIONSHIP BETWEEN HORIZ DR, HORIZ BLANK, HORIZ SYNC AND COLOR BURST
EACH HORIZONTAL DIVISION IS EQUAL TO 3½ CYCLES OF 7M
THE PATTERN REPEATS EVERY 455 CYCLES OF 7M
SHADED AREA VOLTAGE DETERMINED BY THE DATA IN RAM



RELATIONSHIP BETWEEN VERTICAL SYNC, VERTICAL BLANK AND VERTICAL DRIVE

EACH HORIZONTAL DIVISION REPRESENTS ONE HORIZONTAL SCAN

1/14/77	N/C
1/27/77	A
3/25/77	B
7/6/77	C

ELECTRICAL SPECIFICATION FOR MIDWAY CUSTOM CIRCUITS

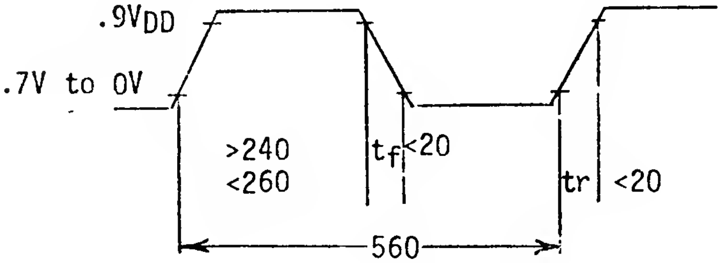
I. GENERAL SYSTEM PARAMETERS

I. A. Power Supplies

1. VDD=+5.0V $\pm$5%
2. VGG=+10.0V $\pm$5%
3. VSS=0.0V

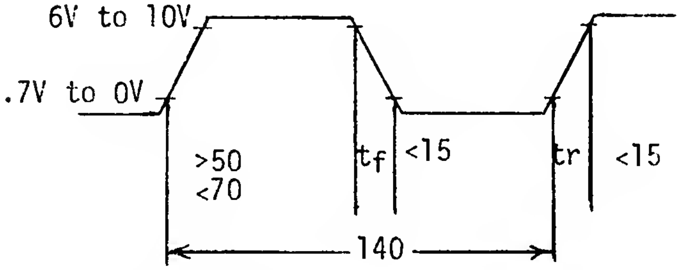
I. B. Timing Signals

1. Ø & $\overline{\text{Ø}}$; Period = 560nsec, High time* 240nsec to 260nsec.
   Ø and $\overline{\text{Ø}}$ have zero level crossover +1 volt -0 volts
   t_r, t_f* less than 20nsec



(Times are in nsec)

2. 7M & $\overline{7M}$; Period = 140nsec, High time$^+$ 50nsec to 70nsec
   7M & 7M have zero level crossover +1 volt -0 volt
   t_r, $t_f$$^+$ less than 15nsec



(Times are in nsec)

Dead time $\leq$ 5nsec
Max C Load = 20pf

+Note
1) High time is time clock at $\geq$.6V.
2) Rise time from zero level to one level.

I. B. (Continued)

*Note:

1. High time is time between 50% points.

2. Clock signals are generated by low power Shottky Logic (series 74LS). Full level swing on clock signals to be achieved through external resistor to V_{DD}. Zero level .7V to 0V.

3. Rise time from zero level to .9V_{DD}.

I. C. <u>Z80 Data Bus (MUXDØ-MUXD7)</u>

1. Z80 Data Bus interface requires a three-state output/input buffer. The three states are defined below.

2. Logic 0: .5V + noise generated by chip, noise for address chip is .15V @ -430µA

3. Logic 1: 2.7V @ +70µA

4. High Impedance: Leakage at either logic 0 or 1 to be less than 5µA.

5. Transient Response: Transition from High Impedance to 0 or 1 will be complete within 442nsec of the 90% point of $\overline{\emptyset}$ of the last wait state of input cycle or 442nsec of the 90% point of Ø of the second wait state of the interrupt acknowledge cycle. The maximum load will be 80pf. This includes 14pfd for two custom chips.

6. Exception: The path through the Data chip connecting the RAM bus with the Z80 bus shall introduce a maximum of 160nsec of delay.

7. The low address byte will be valid on the Z80 Data Bus at least 62nsec before $\overline{\emptyset}$. The high address byte will be valid at least 79nsec before Ø. The data byte will be valid 55nsec before $\overline{\emptyset}$.

I. D. RAM Data Bus (MD0-MD7) - Home Game

1. The RAM Data Bus will require three state logic buffers.
2. Logic 0: .5V @ -25μA
3. Logic 1: 2.7V @ +25μA
4. High Impedance: 5μA maximum leakage at either logic 0 or 1.
5. Transient Response: The outputs shall transition from High Impedance to 0 or 1 within 120nsec of 7M. The outputs shall transition from 1 or 0 to high impedance within 20nsec of 7M. Maximum load will be 20pf.

I. E. RAM Data Bus (MD0-MD7) - Commercial Game

1. The RAM Data Bus will require three state logic buffers.
2. Logic 0: .5V @ -200μA
3. Logic 1: 2.7V @ +25μA
4. High Impedance: 5μA maximum leakage of either logic 0 or 1.
5. Transient Response: The output shall transition from High Impedance to 0 or 1 within 120nsec of 7M. The output shall transition from 1 or 0 to High Impedance within 2nsec of 7M. Maximum load will be 10pf.

I. F. Ambient operating temperature ≥ 0°C, ≤ 55°C.

I. G. Storage temperature ≥ -65°C, ≤ 150°C.

I. H. Packing 40 pin plastic.

II. CUSTOM CIRCUIT SPECIFICATION

This specification defines the terminal characteristics for each of the custom circuits. These specifications shall take precedence in case of conflict. All Ø references refer to the Ø and $\overline{Ø}$ inputs to the address and I/O chip.

II. A. Data Chip

1. Input Pin List

Input Pin List	V0 (V)	V1 (V)	td (Low)[1] (nsec)	t_d (High)[1] (nsec)	Ref.
$\overline{\text{MREQ}}$	.5	2.45	132	.6	7M
$\overline{\text{RD}}$	.5	2.45	12	6	7M
$\overline{\text{IORQ}}$	.5	2.45	112	126	7M
7M	See Section I.B.				
$\overline{\text{7M}}$	"	"			
$\overline{\text{WRCTL}}$	.5	3.1	82	82	7M
$\overline{\text{M1}}$	.5	2.45	12	82	7M
LTCHDO	.5	3.1	120	120	7M
Serial 0	.5	2.45	30	30	7M
Serial 1	.5	2.45	30	30	7M

2. Power Supplies

See Section I. A.

3. Bus Connections

MXD0	See Z80 Data Bus Spec.	Section I.C.
MXD1	"	"
MXD2	"	"
MXD3	"	"
MXD4	"	"
MXD5	"	"
MXD6	"	"
MXD7	"	"
MD0	See RAM Data Bus Spec	Section I.D.
MD1	"	"
MD2	"	"
MD3	"	"
MD4	"	"
MD5	"	"
MD6	"	"
MD7	"	"

4. Outputs	VO	IO	V1	I1	CAP	t_p	Ref.
	(V)	(μA)	(V)	(μA)	(pf)	(nsec)	
VIDEO*	*				10	100	7M
R-Y*	*				10	600	
B-Y*	*				10	600	
HORIZ DR	Note 4	400	2.7	20	20	20	7M
VERT DR	Note 4	400	2.7	20	20	20	7M
2.5V[6]	--	--	--	--	--	DC	
Ø	Note 4	400	2.7	20	10	100	7M
$\overline{\text{PXCLK}}$	Note 4	400	2.7	20	10	100	7M
MC0	Note 4	400	2.7	20	10	120	7M
MC1	Note 4	400	2.7	20	10	120	7M
$\overline{\text{DATEN}}$	Note 4	400	2.7	20	10	90	7M

*Video, R-Y, B-Y are analog outputs at 140nsec rate. Video, must switch from 10% to 90% of blank to white in 140nsec. R-Y and B-Y transitions not to exceed .6μsec.

1 t_d (Low) and t_d (High) is maximum time in nsec except where a minimum is shown.
2 For $\overline{\text{IORQ}}$ Ref. to $\overline{\text{Ø}}$ t_d (Low)=132nsec t_d (High)=6nsec.
3 Serial 0 and Serial 1 will operate at 7MHz.
4 .5V + noise generated by chip.
5 Tap on both resistor chains for a capacitor. Will become test input with voltage applied > 8V.
6 The Z80 Ø is generated by this signal with a clock driver which introduces a delay of <20nsec.

II. B. I/O Chip

1. Input Pin List

Input Pin List	V0	V1	Ref	t_d (High) (nsec)	t_d (Low) (nsec)
Reset	.5	2.45			
MONOS	Note 1				
$\overline{\text{RD}}$	.5	2.45	Ø or $\overline{\text{Ø}}$	166	172 Ø or $\overline{\text{Ø}}$
$\overline{\text{IORQ}}$	.5	2.45	Ø^6	146 $\overline{\text{Ø}}$	132 Ø
Ø	See Section I.B.				
$\overline{\text{Ø}}$	"	"	"		
SIØ	.5	3.3			Note 3
SI1	.5	3.3			Note 3
SI2	.5	3.3			Note 3
SI3	.5	3.3			Note 3
SI4	.5	3.3			Note 3
SI5	.5	3.3			Note 3
SI6	.5	3.3			Note 3
SI7	.5	3.3			Note 3
TEST	.5	5.0			DC

2. Power Supplies

See Section I.A.

3. Bus Connections

MUXD0	See Z80 Data Bus Spec	Section I.C.
MUXD1	"	"
MUXD2	"	"
MUXD3	"	"
MUXD4	"	"
MUXD5	"	"
MUXD6	"	"
MUXD7	"	"

4. Outputs

Outputs		V0 (V)	I0 (µA)	V1 (V)	I1 (µA)
Audio	Note 4	Fmax - 20KHz			
Discharge	Note 5	.5V	4V		
SOØ	Note 3	Note 7	200	4V	1650
SO1	Note 3	Note 7	200	4V	1650
SO2	Note 3	Note 7	200	4V	1650
SO3	Note 3	Note 7	200	4V	1650
SO4	Note 3	Note 7	200	4V	1650
SO5	Note 3	Note 7	200	4V	1650
SO6	Note 3	Note 7	200	4V	1650
SO7	Note 3	Note 7	200	4V	1650
POT Ø	Note 2		5	V_{DD}-.5	50
POT 1	Note 2		5	V_{DD}-.5	50
POT 2	Note 2		5	V_{DD}-.5	50
POT 3	Note 2		5	V_{DD}-.5	50

Note 1 MONOS triggers at 2.1 volts ±2% ± noise voltage when the supply is 5.25V.

Note 2 Open source-Voltage measured with 0.2ma.

Note 3 Time from load of address into microcycle register to date valid on MUX data bus from SI inputs (data path through address decoder, out on SO outputs, through closed switch and isolation diode, into SI input to MUX Data Bus) shall be 2μsec max. Drop of isolation diode will be 0.7V max. SO must drive 2kΩ in the high level. Max C load of SO shall be 300 pf. SI input shall have kill device enabled by $\overline{\text{INPUT}}$.

Note 4 Audio voltage oscillates between 0V and one of the following voltages; .33, .67, 1.00, 1.33, 1.67, 2.00, 2.33, 2.67, 3.00, 3.33, 3.67, 4.00, 4.33, 4.67 and 5.00. These voltages should be ±6%. The load shall be 1000pf and 100kΩ.

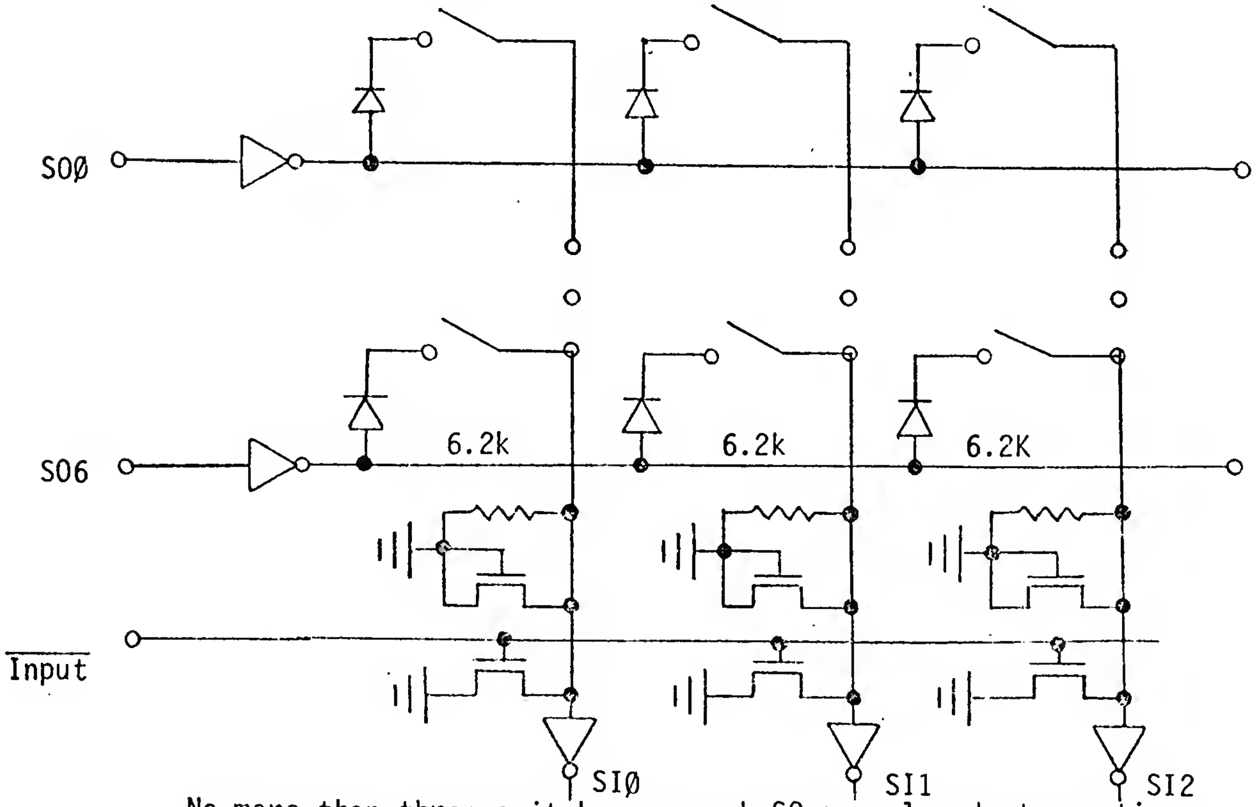
Note 5 Discharge is open drain to V_{SS}. Discharges .01μfd capacitor to .2V in 144μsec.

Note 6 For $\overline{\text{IOREQ}}$ Ref. to $\bar{\emptyset}$ t_d (Low)=152nsec t_d (High)=166nsec.

Note 7 .5V + noise generated by I/O chip.

Miscellaneous Timing

Time for MO Adder - 2∅ max



No more than three switches on each SO are closed at one time.

II. C. Address Chip

1. Input Pin List

Input Pin List	VO (V)	V1 (V)	t_{pd} (Low) (nsec)	t_{pd} (High) (nsec)	REF
$\overline{\text{RFSH}}$	.5	2.45	222 Ø	216	Ø
$\overline{\text{MREQ}}$	.5	2.45	152 $\overline{\emptyset}$	166	Ø or $\overline{\emptyset}$
$\overline{\text{RD}}$	.5	2.45	172 Ø or $\overline{\emptyset}$	166	Ø or $\overline{\emptyset}$
$\overline{\text{MI}}$	.5	2.45	176 Ø	242	Ø
A12[1]	.5	2.45			Ø
A13[1]	.5	2.45			Ø
A14[1]	.5	2.45			Ø
A15[1]	.5	2.45			Ø
$\overline{\text{IORQ}}$	.5	2.45	132 Ø	146	$\overline{\emptyset}$[2]
$\overline{\text{LIGHT PEN}}$	.5	2.45	Asyn		
$\overline{\text{TEST}}$	.5	5.0	DC		
HORIZ. DR.	.5	2.45	Note 3		$\overline{\emptyset}$
VERT. DR.	.5	2.45	Note 4		Ø
Ø	See Section I.B.				
$\overline{\emptyset}$	"	"	"		

2. Power Supplies

See Section I.A.

3. Bus Connections

MXD0	See Z80 Data Bus Spec Section I.E.
MXD1	" "
MXD2	" "
MXD3	" "
MXD4	" "
MXD5	" "
MXD6	" "
MXD7	" "

4. Outputs

Outputs	VO (V)	IO (μA)	V1 (V)	I1 (μA)	CAP (pf)	t_{pd}(Low) (nsec)	t_{pd}(High) (nsec)	REF
LATCHDO	Note 7	Note 6	3.1	Note 6	10	280	140	$\overline{\emptyset}$[5]
$\overline{\text{WAIT}}$	" "	400	2.4	20	25	490	490	$\overline{\emptyset}$
$\overline{\text{MA0-MA5}}$	" "	400	2.4	20	20	242	240	$\overline{\emptyset}$ or Ø
$\overline{\text{INT}}$	" "	400	2.4	20	25	490	572	$\overline{\emptyset}$
$\overline{\text{RAS0-RAS3}}$	" "	400	2.4	20	20	382	382	$\overline{\emptyset}$
$\overline{\text{WRCTL}}$	" "	Note 6	3.1	Note 6	10	382	382	$\overline{\emptyset}$

1. Time from High Impedance to 1 or 0 is 200nsec. (from $Ø_1$ of T_1)
2. For $\overline{\text{IORQ}}$ Ref to $\overline{\emptyset}$ t_d (Low)=152nsec t_d (High)=166nsec. Ø
3. Horizontal Drive time from low to high is 40nsec after $\overline{\emptyset}$.
   Time from high to low is 100nsec before rising edge of Ø.
4. Vertical Drive will transition from low to high 40nsec after falling edge of Ø. Its width will be 2.1 μsec max. 1.54μsec min. It will go from high to low 100nsec before falling edge of Ø.
5. Reference t_{pd} (High) is Ø.
6. MOS to MOS signal.
7. .5V + noise generated by Address Chip (.15V) = .65V

III. I/O MODE DECODE

I/O Parts

HEX	Out	Input
0	Color Ø Right	
1	" 1 "	
2	" 2 "	
3	" 3 "	
4	" 0 Left	
5	" 1 "	
6	" 2 "	
7	" 3 "	
8	Consumer/Commercial	Intercept Feedback
9	Horiz Color Bndry	
A	Vertical Blank	
B	Color Block TX	
C	Magic Reg	
D	Interrupt Feedback	
E	Interrupt Mode	Vertical Addr Feedback
F	Interrupt Line	Horizontal Addr Feedback
10	Tone Master OSC	SW Bank 0
11	Tone A	1
12	" B	2
13	" C	3
14	Tremello	4
15	Tone C Volume	5
16	Tone A,B Volume	6
17	Noise Volume	7
18	Sound Block TX	
19		
1A		
1B		
1C		POT 0
1D		" 1
1E		" 2
1F		" 3
20		
21		
22		
23		
24		
.		
.		
2F		

Software and Hardware for the Bally Arcade - A Technical Description

A Dave Nutting Associates Design

Bally Arcade

8K ROM Source Listing

	Name	Pages	ROM Memory
1)	Home Video Game Equates	2 - 15	
2)	System Routines	16 - 94	$0000
3)	Scribbling	1 - 17	$0E19
4)	Calculator	1 - 20	$1020
5)	Checkmate	1 - 30	$1328
6)	Gun Fight	1 - 46	$17DE

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

                 30    ; ***************************
                 31    ; * HOME VIDEO GAME EQUATES *
                 32    ; ***************************
                 33    ;
                 34    ; ASSEMBLY CONTROL
                 35    ;
>0001            36  XPNDON  EQU  1            ; ** SET TO 1 WHEN HARDWARE EXP
>0001            37  NWHDWR  EQU  1            ; ** SET TO 1 WHEN NEW HARDWARE
                 38    ;
                 39    ; GENERAL GOODIES
>4000            40  NORMEM  EQU  4000H
>2000            41  FIRSTC  EQU  2000H        ; FIRST ADDRESS IN CASSETTE
>0000            42  SCREEN  EQU  0
>0028            43  BYTEPL  EQU  40           ; BYTES PER LINE
>00A0            44  BITSPL  EQU  160          ; BITS PER LINE
                 45  ; STUFF IN SYSTEM DOPE VECTOR
>0200            46  STIMER  EQU  200H         ; SECONDS AND GAME TIME,MUSIC
>0203            47  CTIMER  EQU  203H         ; CUSTOM TIMERS
>0206            48  FNTSYS  EQU  206H         ; SYSTEM FONT DESCRIPTOR
>020D            49  FNTSML  EQU  20DH         ; SMALL FONT DESCRIPTOR
>0214            50  ALKEYS  EQU  214H         ; KEYMASK OF ALL KEYS
>0218            51  MENUST  EQU  218H         ; HEAD OF ONBOARD MENU
>021E            52  MXSCR   EQU  21EH         ; ADDRESS OF 'MAX SCORE'
>0228            53  NOPLAY  EQU  228H         ; ADDRESS OF '# OF PLAYERS'
>0235            54  NOGAME  EQU  235H         ; ADDRESS OF '# OF GAMES'
                 55    ; BITS IN PROCESSOR FLAG BYTE
>0007            56  PSWSGN  EQU  7            ; SIGN BIT
>0006            57  PSWZRO  EQU  6            ; ZERO BIT
>0002            58  PSWPV   EQU  2            ; PARITY        OVERFLOW
>0000            59  PSWCY   EQU  0            ; CARRY
                 60    ; BITS IN GAME STATUS BYTE
>0000            61  GSBTIM  EQU  0
>0001            62  GSBSCR  EQU  1
>0007            63  GSBEND  EQU  7
                 64    ; STANDARD VECTOR DISPLACEMENTS AND BITS
>0000            65  VBMR    EQU  0            ; MAGIC REGISTER
>0001            66  VBSTAT  EQU  1            ; STATUS
>0002           '67  VBTIMB  EQU  2            ; TIME BASE
>0003            68  VBDXL   EQU  3            ; DELTA X LO
>0004            69  VBDXH   EQU  4            ; DELTA X HI
>0005            70  VBXL    EQU  5            ; X COORD LO
>0006            71  VBXH    EQU  6            ; X COORD HI
>0007            72  VBXCHK  EQU  7            ; X CHECK FLAGS
>0008            73  VBDYL   EQU  8            ; DELTA Y LO
>0009            74  VBDYH   EQU  09H          ; DELTA Y HI
>000A            75  VBYL    EQU  0AH          ; Y COORD LO
>000B            76  VBYH    EQU  0BH          ; Y COORD HI
>000C            77  VBYCHK  EQU  0CH          ; Y CHECK FLAGS
>000D            78  VBOAL   EQU  0DH          ; OLD ADDRESS L. O.
>000E            79  VBOAH   EQU  0EH          ; OLD ADDRESS H. O.
                 80     ; DISPLACEMENTS FROM START OF COORDINATE AREA
>0000            81  VBDCL   EQU  0            ; LO DELTA
>0001            82  VBDCH   EQU  1            ; HI DELTA
>0002            83  VBCL    EQU  2            ; LO COORD
>0003            84  VBCH    EQU  3            ; HI COORD
>0004            85  VBCCHK  EQU  4            ; CHECK BITS
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

                86    ; BITS IN STATUS BYTE
>0007           87  VBSACT   EQU  7              ; VECTOR ACTIVE STATUS
>0006           88  VBBLNK   EQU  6              ; BLANK STATUS
                89    ; BITS IN CHECK BIT MASK
>0000           90  VBCLMT   EQU  0              ; DO LIMIT CHECKING
>0001           91  VBCREV   EQU  1              ; REVERSE DELTA ON LIMIT ATTAIN
>0003           92  VBCLAT   EQU  3              ; COORDINATE IS AT LIMIT
                93   ; FONT TABLE DISPLACEMENTS FOR NEW CHARACTER DISPLAY ROU
>0000           94  FTBASE   EQU  0              ; BASE CHARACTER
>0001           95  FTFSX    EQU  1              ; X FRAME SIZE
>0002           96  FTFSY    EQU  2              ; Y FRAME SIZE
>0003           97  FTBYTE   EQU  3              ; X SIZE OF CHAR IN BYTES
>0004           98  FTYSIZ   EQU  4              ; Y SIZE IN BITS
>0005           99  FTPTL    EQU  5              ; PATTERN TABLE ADDRESS LO
>0006          100  FTPTH    EQU  6              ; PATTERN TABLE ADDRESS HI
               101    ; BITS FOR MAGIC REGISTER       WRITE OPTION BYTE
>0006          102  MRFLOP   EQU  6              ; WRITE WITH FLOP
>0005          103  MRXOR    EQU  5              ; WRITE WITH EXCLUSIVE OR
>0004          104  MROR     EQU  4              ; WRITE WITH OR
>0003          105  MRXPND   EQU  3              ; WRITE WITH EXPAND
>0002          106  MRROT    EQU  2              ; WRITE WITH ROTATE
>0003          107  MRSHFT   EQU  03H            ; MASK OF SHIFT AMOUNT
               108    ; BITS OF CONTROL HANDLE INPUT PORT
>0004          109  CHTRIG   EQU  4              ; TRIGGER
>0003          110  CHRIGH   EQU  3              ; JOYSTICK RIGHT
>0002          111  CHLEFT   EQU  2              ; JOYSTICK LEFT
>0001          112  CHDOWN   EQU  1              ; DOWN
>0000          113  CHUP     EQU  0              ; UP
               114    ; CONTEXT BLOCK REGISTER DISPLACEMENTS
>0000          115  CBIYL    EQU  0              ; IY
>0001          116  CBIYH    EQU  1
>0002          117  CBIXL    EQU  2              ; IX
>0003          118  CBIXH    EQU  3
>0004          119  CBE      EQU  4              ; DE
>0005          120  CBD      EQU  5
>0006          121  CBC      EQU  6              ; BC
>0007          122  CBB      EQU  7
>0008          123  CBFLAG   EQU  8              ; AF
>0009          124  CBA      EQU  9
>000A          125  CBL      EQU  0AH            ; HL
>000B          126  CBH      EQU  0BH
               127  ; SENTRY RETURN CODE EQUATES:
>0000          128  SNUL     EQU  0              ; NOTHING HAPPENED
>0001          129  SCT0     EQU  1              ; COUNTER-TIMER 1 THRU 8
>0002          130  SCT1     EQU  2
>0003          131  SCT2     EQU  3
>0004          132  SCT3     EQU  4
>0005          133  SCT4     EQU  5
>0006          134  SCT5     EQU  6
>0007          135  SCT6     EQU  7
>0008          136  SCT7     EQU  8
>0009          137  SF0      EQU  9              ; FLAG BIT 0
>000A          138  SF1      EQU  0AH
>000B          139  SF2      EQU  0BH
>000C          140  SF3      EQU  0CH
>000D          141  SF4      EQU  0DH
>000E          142  SF5      EQU  0EH
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

>000F           143  SF6      EQU  0FH
>0010           144  SF7      EQU  10H
>0011           145  SSEC     EQU  11H            ; SECONDS TIMER HAS COUNTED DOW
>0013           146  SKYD     EQU  13H            ; KEY IS DOWN
>0012           147  SKYU     EQU  12H            ; YES IS UP
>001C           148  SPO      EQU  1CH            ; POT O
>001D           149  SP1      EQU  1DH            ; POT 1
>001E           150  SP2      EQU  1EH            ; POT 2
>001F           151  SP3      EQU  1FH            ; POT 3
>0014           152  STO      EQU  14H            ; TRIGGER O
>0015           153  SJO      EQU  15H            ; JOYSTICK O
>0016           154  ST1      EQU  16H            ; SIMILARLY FOR 1-3
>0017           155  SJ1      EQU  17H
>0018           156  ST2      EQU  18H
>0019           157  SJ2      EQU  19H
>001A           158  ST3      EQU  1AH
>001B           159  SJ3      EQU  1BH
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

                161    ; ********************************
                162    ; * HOME VIDEO GAME PORT EQUATES *
                163    ; ********************************
                164    ; OUTPUT PORTS FOR VIRTUAL COLOR
>0000           165  COLOR    EQU  0              ; COLOR 0 RIGHT
>0001           166  COL1R    EQU  1              ; COLOR 1 RIGHT
>0002           167  COL2R    EQU  2              ; COLOR 2 RIGHT
>0003           168  COL3R    EQU  3              ; COLOR 3 RIGHT
>0004           169  COLOL    EQU  4              ; COLOR 0 LEFT
>0005           170  COL1L    EQU  5              ; COLOR 1 LEFT
>0006           171  COL2L    EQU  6              ; COLOR 2 LEFT
>0007           172  COL3L    EQU  7              ; COLOR 3 LEFT
>000B           173  COLBX    EQU  0BH            ; COLOR BLOCK OUTPUT PORT
>0009           174  HORCB    EQU  9              ; HORIZONTAL COLOR BOUNDARY
>000A           175  VERBL    EQU  0AH            ; VERTICAL BLANKING LINE
                176    ; OUTPUT PORTS FOR MUSIC AND SOUNDS
>0010           177  TONMO    EQU  10H            ; TONE MASTER OSCILLATOR
>0011           178  TONEA    EQU  11H            ; TONE A OSC.
>0012           179  TONEB    EQU  12H            ; TONE B OSC.
>0013           180  TONEC    EQU  13H            ; TONE C OSC.
>0014           181  VIBRA    EQU  14H            ; VIBRATO
>0016           182  VOLAB    EQU  16H            ; TONES A,B VOLUME
>0015           183  VOLC     EQU  15H            ; TONE C VOLUME
>0017           184  VOLN     EQU  17H            ; NOISE VOLUME
>0018           185  SNDBX    EQU  18H            ; SOUND BLOCK OUTPUT PORT
                186    ; INTERRUPT AND CONTROL OUTPUT PORTS
>000D           187  INFBK    EQU  0DH            ; INTERRUPT FEEDBACK
>000E           188  INMOD    EQU  0EH            ; INTERRUPT MODE
>000F           189  INLIN    EQU  0FH            ; INTERRUPT LINE
>0008           190  CONCM    EQU  8              ; CONSUMER      COMMERCIAL
>000C           191  MAGIC    EQU  0CH            ; MAGIC REGISTER
>0019           192  XPAND    EQU  19H            ; EXPANDER PIXEL DEFINITION POR
                193    ; INTERRUPT AND INTERCEPT INPUT PORTS
>0008           194  INTST    EQU  8              ; INTERCEPT STATUS
>000E           195  VERAF    EQU  0EH            ; VERTICAL ADDRESS FEEDBACK
>000F           196  HORAF    EQU  0FH            ; HORIZONTAL ADDRESS FEEDBACK
                197    ; HAND CONTROLS INPUT PORTS
>0010           198  SW0      EQU  10H            ; PLAYER 0 HAND CONTROL
>0011           199  SW1      EQU  11H            ; PLAYER 1 HAND CONTROL
>0012           200  SW2      EQU  12H            ; PLAYER 2 HAND CONTROL
>0013           201  SW3      EQU  13H            ; PLAYER 3 HAND CONTROL
>001C           202  POT0     EQU  1CH            ; PLAYER 0 POT
>001D           203  POT1     EQU  1DH            ; PLAYER 1 POT
>001E           204  POT2     EQU  1EH            ; PLAYER 2 POT
>001F           205  POT3     EQU  1FH            ; PLAYER 3 POT
                206    ; KEYBOARD INPUT PORTS
>0014           207  KEY0     EQU  14H            ; KEYBOARD COLUMN 0
>0015           208  KEY1     EQU  15H            ; KEYBOARD COLUMN 1
>0016           209  KEY2     EQU  16H            ; KEYBOARD COLUMN 2
>0017           210  KEY3     EQU  17H            ; KEYBOARD COLUMN 3
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                212    ; *************************************
                213    ; * HOME VIDEO GAME SYSTEM CALL INDEXES *
                214    ; *************************************
                215    ; USER PROGRAM INTERFACE
>0000           216  UPISTR   EQU  0
>0000           217  INTPC    EQU  UPISTR        ; INTERPRET WITH CONTEXT CREATE
>0002           218  XINTC    EQU  INTPC+2       ; EXIT INTERPRETER WITH CONTEXT
>0004           219  RCALL    EQU  XINTC+2       ; CALL ASM LANGUAGE SUBROUTINE
>0006           220  MCALL    EQU  RCALL+2       ; CALL INTERPRETER SUBROUTINE
>0008           221  MRET     EQU  MCALL+2       ; RETURN FROM INTERPRETER SUBRO
>000A           222  MJUMP    EQU  MRET+2        ; MACRO JUMP
>000C           223  SUCK     EQU  MJUMP+2       ; SUCK INLINE ARGS INTO CB
                224    ; SCHEDULER ROUTINES
>000C           225  SCHEDR   EQU  SUCK
>000E           226  ACTINT   EQU  SCHEDR+2      ; SET SUB TIMER
>0010           227  DECCTS   EQU  ACTINT+2      ; DEC CT'S UNDER MASK
                228    ; MUSIC AND SOUNDS
>0012           229  MUZAK    EQU  DECCTS+2
>0012           230  BMUSIC   EQU  MUZAK         ; BEGIN PLAYING MUSIC
>0014           231  EMUSIC   EQU  BMUSIC+2      ; STOP PLAYING MUSIC
                232    ; SCREEN HANDLER ROUTINES
>0016           233  SCRSTR   EQU  EMUSIC+2
>0016           234  SETOUT   EQU  SCRSTR        ; SET SCREEN SIZE
>0018           235  COLSET   EQU  SETOUT+2      ; SET COLORS
>001A           236  FILL     EQU  COLSET+2      ; FILL MEMORY WITH CONSTANT DAT
>001C           237  RECTAN   EQU  FILL+2        ; PAINT RECTANGLE
>001E           238  VWRITR   EQU  RECTAN+2      ; WRITE RELATIVE FROM VECTOR
>0020           239  WRITR    EQU  VWRITR+2      ; WRITE RELATIVE
>0022           240  WRITP    EQU  WRITR+2       ; WRITE WITH PATTERN SIZE LOOKU
>0024           241  WRIT     EQU  WRITP+2       ; WRITE WITH SIZES PROVIDED
>0026           242  WRITA    EQU  WRIT+2        ; WRITE ABSOLUTE
>0028           243  VBLANK   EQU  WRITA+2       ; BLANK AREA FROM VECTOR
>002A           244  BLANK    EQU  VBLANK+2      ; BLANK AREA
>002C           245  SAVE     EQU  BLANK+2       ; SAVE AREA
>002E           246  RESTOR   EQU  SAVE+2        ; RESTORE AREA
>0030           247  SCROLL   EQU  RESTOR+2      ; SCROLL AREA OF SCREEN
                248    ;
>0032           249  CHRDIS   EQU  SCROLL+2      ; NEW DISPLAY CHARACTER
>0034           250  STRDIS   EQU  CHRDIS+2      ; NEW DISPLAY STRING
>0036           251  DISNUM   EQU  STRDIS+2      ; DISPLAY NUMBER
                252    ;
>0038           253  RELABS   EQU  DISNUM+2      ; RELATIVE TO ABSOLUTE CONVERSI
>003A           254  RELAB1   EQU  RELABS+2      ; NONMAGIC RELABS
>003C           255  VECTC    EQU  RELAB1+2      ; VECTOR SINGLE COORDINATE
>003E           256  VECT     EQU  VECTC+2       ; VECTOR COORDINATE PAIR
                257    ; HUMAN INTERFACE ROUTINES
>0040           258  HUMANR   EQU  VECT+2
>0040           259  KCTASC   EQU  HUMANR        ; KEY CODE TO ASCII
>0042           260  SENTRY   EQU  KCTASC+2      ; SENSE TRANSITION
>0044           261  DOIT     EQU  SENTRY+2      ; BRANCH TO TRANSITION HANDLER
>0046           262  DOITB    EQU  DOIT+2        ; USE B INSTEAD OF A
>0048           263  PIZBRK   EQU  DOITB+2       ; TAKE A BREAK
>004A           264  MENU     EQU  PIZBRK+2      ; DISPLAY A MENU
>004C           265  GETPAR   EQU  MENU+2        ; GET GAME PARAMETER FROM USER
>004E           266  GETNUM   EQU  GETPAR+2      ; GET NUMBER FROM USER
>0050           267  PAWS     EQU  GETNUM+2      ; PAUSE
```

```
 ADDR OBJECT    STMT  LABEL    OPCD OPERAND      COMMENT

>0052            268  DISTIM   EQU  PAWS+2       ; DISPLAY TIME
>0054            269  INCSCR   EQU  DISTIM+2     ; INC SCORE
                 270   ; MATH ROUTINES
>0056            271  MATH     EQU  INCSCR+2
>0056            272  INDEXN   EQU  MATH         ; INDEX NIBBLE
>0058            273  STOREN   EQU  INDEXN+2
>005A            274  INDEXW   EQU  STOREN+2     ; INDEX WORD
>005C            275  INDEXB   EQU  INDEXW+2     ; INDEX BYTE
>005E            276  MOVE     EQU  INDEXB+2     ; BLOCK TRANSFER
>0060            277  SHIFTU   EQU  MOVE+2       ; SHIFT UP A DIGIT
>0062            278  BCDADD   EQU  SHIFTU+2     ; BCD ADD
>0064            279  BCDSUB   EQU  BCDADD+2     ; BCD SUBTRACT
>0066            280  BCDMUL   EQU  BCDSUB+2     ; BCD MULTIPLY
>0068            281  BCDDIV   EQU  BCDMUL+2     ; BCD DIVIDE
>006A            282  BCDCHS   EQU  BCDDIV+2     ; BCD CHANGE SIGN
>006C            283  BCDNEG   EQU  BCDCHS+2     ; BCD NEGATE
>006E            284  DADD     EQU  BCDNEG+2     ; DECIMAL ADD
>0070            285  DSMG     EQU  DADD+2       ; CONVERT TO SIGN MAGNITUDE
>0072            286  DABS     EQU  DSMG+2       ; DECIMAL ABSOLUTE VALUE
>0074            287  NEGT     EQU  DABS+2       ; NEGATE
>0076            288  RANGED   EQU  NEGT+2       ; RANGED RANDOM NUMBER
>0078            289  QUIT     EQU  RANGED+2     ; QUIT CASSETTE EXECUTION
>007A            290  SETB     EQU  QUIT+2       ; SET BYTE
>007C            291  SETW     EQU  SETB+2       ; SET WORD
>007E            292  MSKTD    EQU  SETW+2       ; MASK TO DELTAS
```

```
               294    ; **********
               295    ; * MACROS *
               296    ; **********
               297    ; MACROS TO DEFINE PATTERNS
               298  DEF2    MACR #AA,#AB
               299          DEFB #AA
               300          DEFB #AB
               301          ENDM
               302  DEF3    MACR #BA,#BB,#BC
               303          DEFB #BA
               304          DEFB #BB
               305          DEFB #BC
               306          ENDM
               307  DEF4    MACR #CA,#CB,#CC,#CD
               308          DEFB #CA
               309          DEFB #CB
               310          DEFB #CC
               311          DEFB #CD
               312          ENDM
               313  DEF5    MACR #DA,#DB,#DC,#DD,#DE
               314          DEFB #DA
               315          DEFB #DB
               316          DEFB #DC
               317          DEFB #DD
               318          DEFB #DE
               319          ENDM
               320  DEF6    MACR #EA,#EB,#EC,#ED,#EE,#EF
               321          DEFB #EA
               322          DEFB #EB
               323          DEFB #EC
               324          DEFB #ED
               325          DEFB #EE
               326          DEFB #EF
               327          ENDM
               328  DEF8    MACR #GA,#GB,#GC,#GD,#GE,#GF,#GG,#GH
               329          DEFB #GA
               330          DEFB #GB
               331          DEFB #GC
               332          DEFB #GD
               333          DEFB #GE
               334          DEFB #GF
               335          DEFB #GG
               336          DEFB #GH
               337          ENDM
               338    ; MACROS TO COMPUTE CONSTANT SCREEN ADDRESSES
               339  XYRELL  MACR #R,#X,#Y      ;RELATIVE LOAD
               340          LD   #R,.RES.(#Y).SHL.8+(#X)
               341          ENDM
               342   ; MACRO TO GENERATE SYSTEM CALL
               343  SYSTEM  MACR #NUMBA
               344          RST  56
               345          DEFB #NUMBA
               346          IF   #NUMBA.EQ.INTPC
               347  INTP@   DEFL 1
               348          ENDIF
               349          ENDM
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               350   ; MACRO TO GENERATE SYSTEM CALL WITH SUCK OPTION ON
               351  SYSSUK   MACR #UMBA
               352           RST  56
               353           DEFB #UMBA+1
               354           IF   #UMBA.EQ.INTPC
               355  INTP@    DEFL 1
               356           ENDIF
               357           ENDM
               358    ; MACROS TO GENERATE MACRO INSTRUCTION CALLS
               359    ; FILL SCREEN WITH CONSTANT DATA
               360  FILL?    MACR #START,#NBYTES,#DATA
               361           DEFB FILL+1
               362           DEFW #START
               363           DEFW #NBYTES
               364           DEFB #DATA
               365           ENDM
               366    ; EXIT INTERPRETER WITH CONTEXT RESTORE
               367  EXIT     MACR
               368           DEFB XINTC
               369  INTP@    DEFL 0
               370           ENDM
               371    ;INTERPRET WITH INLINE SUCK
               372  DO       MACR #CID
               373           DEFB #CID+1
               374           ENDM
               375    ;INTERPRET WITHOUT INLINE SUCK
               376  DONT     MACR #CID
               377           DEFB #CID
               378           ENDM
               379    ;MACRO CALL FROM DOIT TABLE
>00C0          380  END      EQU  0C0H
               381  MC       MACR #A,#B,#E
               382           DEFB #A+80H
               383           DEFW #B
               384           IF   0#E
               385           DEFB 0#E
               386           ENDIF
               387           ENDM
               388    ;REAL CALL FROM DOIT TABLE
               389  RC       MACR #A,#B,#E
               390           DEFB #A+40H
               391           DEFW #B
               392           IF   0#E
               393           DEFB 0#E
               394           ENDIF
               395           ENDM
               396    ;REAL JUMP FROM DOIT TABLE
               397  JMP      MACR #A,#B,#E
               398           DEFB #A
               399           DEFW #B
               400           IF   0#E
               401           DEFB 0#E
               402           ENDIF
               403           ENDM
               404    ;DISPLAY A STRING
               405  TEXT     MACR #A,#B,#C,#D
               406           DEFB STRDIS+1
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

               407           DEFB #B
               408           DEFB #C
               409           DEFB #D
               410           DEFW #A
               411           ENDM



               413     ;***************
               414     ; MUSIC MACROS
               415     ; NOTE DURATION,FREQ(S)
               416  NOTE1    MACR #DUR,#N1
               417           DEFB #DUR&7FH
               418           DEFB #N1
               419           ENDM
               420  NOTE2    MACR #DUR,#N1,#N2
               421           DEFB #DUR&7FH
               422           DEFB #N1
               423           DEFB #N2
               424           ENDM
               425  NOTE3    MACR #DUR,#N1,#N2,#N3
               426           DEFB #DUR
               427           DEFB #N1
               428           DEFB #N2
               429           DEFB #N3
               430           ENDM
               431  NOTE4    MACR #DUR,#N1,#N2,#N3,#N4
               432           DEFB #DUR
               433           DEFB #N1
               434           DEFB #N2
               435           DEFB #N3
               436           DEFB #N4
               437           ENDM
               438  NOTE5    MACR #DUR,#N1,#N2,#N3,#N4,#N5
               439           DEFB #DUR
               440           DEFB #N1
               441           DEFB #N2
               442           DEFB #N3
               443           DEFB #N4
               444           DEFB #N5
               445           ENDM
               446  MASTER   MACR #OFFSET
               447           DEFB 80H
               448           DEFB #OFFSET
               449           ENDM
               450     ; STUFF OUTPUT PORT#,DATA  OR
               451     ; OUTPUT SNDBX,DATA10,D11,...,DATA17
               452  OUTPUT   MACR #PORT,#D0,#D1,#D2,#D3,#D4,#D5,#D6,#D7
               453           IF   .NOT.(#PORT=18H)
               454           DEFB 80H+(#PORT&7FH)
               455           DEFB #D0
               456           ENDIF
               457           IF   #PORT=18H
               458           DEFB 88H
               459           DEF8 #D7,#D6,#D5,#D4,#D3,#D2,#D1,#D0
```

```
                 460           ENDIF
                 461           ENDM
                 462     ;  SET VOICE BYTE
                 463     ;  THE FORMAT OF THE VOICE BYTE IS
                 464     ;  *I*A*I*B*I*C*V*N*
                 465     ;  WHERE N = LOAD NOISE WITH DATA AT PC AND INC PC
                 466     ;  V = LOAD VIBRATO AND INC PC
                 467     ;  I = INC PC
                 468     ;  A,B,C = LOAD TONE A,B,C WITH DATA AT PC
                 469   VOICES  MACR #MASK
                 470           DEFB 90H
                 471           DEFB #MASK
                 472           ENDM
                 473   ; PUSH NUMBER ONTO STACK
                 474   PUSHN   MACR #NUMB
                 475           DEFB 0A0H+((#NUMB-1).AND.0FH)
                 476           ENDM
                 477     ; SET VOLUMES
                 478   VOLUME  MACR #BA,#MC
                 479           DEFB 0B0H
                 480           DEFB #BA
                 481           DEFB #MC
                 482           ENDM
                 483   ; CALL RELATIVE 0-15 BEYOND SELF+1
                 484   CREL    MACR #BY
                 485           DEFB 0D0H+(#BY.AND.0FH)
                 486           ENDM
                 487   ; DEC STACK TOP AND JNZ
                 488   DSJNZ   MACR #ADD
                 489           DEFB 0C0H
                 490           DEFW #ADD
                 491           ENDM
                 492   ; FLIP LEGATO   STACATO
                 493   LEGSTA  MACR
                 494           DEFB 0E0H
                 495           ENDM
                 496   REST    MACR #TIME
                 497           DEFB 0E1H
                 498           DEFB #TIME
                 499           ENDM
                 500   QUIET   MACR
                 501           DEFB 0F0H
                 502           ENDM
                 503     ; ****************
                 504     ; * MUSIC EQUATES *
                 505     ; ****************
                 506     ; NOTE VALUES
>00FD            507   G0      EQU  253
>00EE            508   GS0     EQU  238
>00E1            509   A0      EQU  225
>00D4            510   AS0     EQU  212
>00C8            511   B0      EQU  200
>00BD            512   C1      EQU  189
>00B2            513   CS1     EQU  178
>00A8            514   D1      EQU  168
>009F            515   DS1     EQU  159
>0096            516   E1      EQU  150
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

>008D           517  F1       EQU  141
>0085           518  FS1      EQU  133
>007E           519  G1       EQU  126
>0077           520  GS1      EQU  119
>0070           521  A1       EQU  112
>006A           522  AS1      EQU  106
>0064           523  B1       EQU  100
>005E           524  C2       EQU  94
>0059           525  CS2      EQU  89
>0054           526  D2       EQU  84
>004F           527  DS2      EQU  79
>004A           528  E2       EQU  74
>0046           529  F2       EQU  70
>0042           530  FS2      EQU  66
>003E           531  G2       EQU  62
>003B           532  GS2      EQU  59
>0037           533  A2       EQU  55
>0034           534  AS2      EQU  52
>0031           535  B2       EQU  49
>002E           536  C3       EQU  46
>002C           537  CS3      EQU  44
>0029           538  D3       EQU  41
>0027           539  DS3      EQU  39
>0025           540  E3       EQU  37
>0022           541  F3       EQU  34
>0020           542  FS3      EQU  32
>001F           543  G3       EQU  31
>001D           544  GS3      EQU  29
>001B           545  A3       EQU  27
>001A           546  AS3      EQU  26
>0018           547  B3       EQU  24
>0017           548  C4       EQU  23
>0015           549  CS4      EQU  21
>0014           550  D4       EQU  20
>0013           551  DS4      EQU  19
>0012           552  E4       EQU  18
>0011           553  F4       EQU  17
>0010           554  FS4      EQU  16
>000F           555  G4       EQU  15
>000E           556  GS4      EQU  14
>000D           557  A4       EQU  13
>000B           558  C5       EQU  11
>000A           559  CS5      EQU  10
>0009           560  DS5      EQU  9
>0008           561  F5       EQU  8
>0007           562  G5       EQU  7
>0006           563  A5       EQU  6
>0005           564  C6       EQU  5
>0004           565  DS6      EQU  4
>0003           566  G6       EQU  3
>0002           567  C7       EQU  2
>0001           568  G7       EQU  1
>0000           569  G8       EQU  0
                570    ; MASTER OSCILATOR OFFSETS
>00FE           571  OB0      EQU  254
>00F1           572  OC0      EQU  241
>00D6           573  OD1      EQU  214
```

```
 ADDR OBJECT   STMT  LABEL    OPCD OPERAND      COMMENT

>00BF           574  OE1      EQU  191
>00B4           575  OF1      EQU  180
>00A0           576  OG1      EQU  160
>008F           577  OA1      EQU  143
>0047           578  OA2      EQU  71
>0023           579  OA3      EQU  35
>0011           580  OA4      EQU  17
>0008           581  OA5      EQU  8
```

```
                583    ; ***************************
                584    ; * SYSTEM RAM MEMORY CELLS *
                585    ; ***************************
>0FFF           586  WASTE    EQU  0FFFH
>0FFF           587  WASTER   EQU  WASTE
                588    ;
                589    ; THE FOLLOWING ORG SHOULD BE SET TO THE VALUE OF
                590    ; THE TAG 'SYSRAM', THIS WILL CAUSE SYSTEM RAM
                591    ; TO RESIDE AT THE HIGHEST POSSIBLE ADDRESS
                592    ;
                593           ORG  4FC8H
 4FC8           594           DEFS 6              ; GOT SOME LEFT STILL
>4FCE           595  BEGRAM   EQU  $
                596    ; USED BY MUSIC PROCESSOR
 4FCE           597  MUZPC:   DEFS 2              ; MUSIC PROGRAM COUNTER
 4FD0           598  MUZSP:   DEFS 2              ; MUSIC STACK POINTER
 4FD2           599  PVOLAB:  DEFS 1              ; PRESET VOLUME FOR TONES A AND
 4FD3           600  PVOLMC:  DEFS 1              ; PRESET VOLUME FOR MASTER OSC
 4FD4           601  VOICES:  DEFS 1              ; MUSIC VOICES
                602    ; COUNTER TIMERS (USED BY DECCTS,ACTINT,CTIMER)
 4FD5           603  CT0:     DEFS 1              ; COUNTER TIMER 0
 4FD6           604  CT1:     DEFS 1              ; 1
 4FD7           605  CT2:     DEFS 1              ; 2
 4FD8           606  CT3:     DEFS 1              ; 3
 4FD9           607  CT4:     DEFS 1              ; 4
 4FDA           608  CT5:     DEFS 1              ; 5
 4FDB           609  CT6:     DEFS 1              ; 6
 4FDC           610  CT7:     DEFS 1              ; 7
                611    ; USED BY SENTRY TO TRACK CONTROLS
 4FDD           612  CNT:     DEFS 1              ;COUNTER UPDATE&NUMBER TRACKING
 4FDE           613  SEMI4S:  DEFS 1              ; FLAG BITS
 4FDF           614  OPOT0:   DEFS 1              ; POT 0 TRACKING
 4FE0           615  OPOT1:   DEFS 1              ; POT 1 TRACKING
 4FE1           616  OPOT2:   DEFS 1              ; POT 2 TRACKING
 4FE2           617  OPOT3:   DEFS 1              ; POT 3 TRACKING
 4FE3           618  KEYSEX:  DEFS 1              ; KEYBOARD TRACKING BYTE
 4FE4           619  OSW0:    DEFS 1              ; SWITCH 0 TRACKING
 4FE5           620  OSW1:    DEFS 1              ; SWITCH 1 TRACKING
 4FE6           621  OSW2:    DEFS 1              ; SWITCH 2 TRACKING
 4FE7           622  OSW3:    DEFS 1              ; SWITCH 3 TRACKING
 4FE8           623  COLLST:  DEFS 2              ; COLOR LIST ADDRESS FOR P. B.  A
                624    ; USED BY STIMER
 4FEA           625  DURAT:   DEFS 1              ; NOTE DURATION
 4FEB           626  TMR60:   DEFS 1              ; SIXTIETHS OF SEC
 4FEC           627  TIMOUT:  DEFS 1              ; BLAKOUT TIMER
 4FED           628  GTSECS:  DEFS 1              ; GAME TIME SECONDS
 4FEE           629  GTMINS:  DEFS 1              ; GAME TIME MINUTES
                630    ; USED BY MENU
 4FEF           631  RANSHT:  DEFS 4              ; RANDOM NUMBER SHIFT REGISTER
 4FF3           632  NUMPLY:  DEFS 1              ; NUMBER OF PLAYERS
 4FF4           633  ENDSCR:  DEFS 3              ; SCORE TO 'PLAY TO'
 4FF7           634  MRLOCK:  DEFS 1              ; MAGIC REGISTER LOCK OUT FLAG
 4FF8           635  GAMSTB:  DEFS 1              ; GAME STATUS BYTE
 4FF9           636  PRIOR:   DEFS 1              ; MUSIC PROTECT FLAG
 4FFA           637  SENFLG:  DEFS 1              ; SENTRY CONTROL SEIZURE FLAG
 4FFB           638  UMARGT:  DEFS 2
 ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

 4FFD            639  USERTB:  DEFS 2
>4FCE            640  SYSRAM   EQU  (5000H-($-BEGRAM+1))
```

```
 ADDR OBJECT     STMT  LABEL    OPCD OPERAND       COMMENT

                  642
                  643           LIST S,X,T,M
                  644           NLIST I
                  645     ; **********
                  646     ; * HVGSYS *
                  647     ; **********

>0008             649  PFUG     EQU   08H             ; POT FUDGE FACTOR
>17DE             650  GFSTRT   EQU   17DEH           ; GUN FIGHT START ADDRESS
>1328             651  CMSTRT   EQU   1328H           ; CHECKMATE START ADDRESS
>1020             652  CALCST   EQU   1020H           ; CALCULATOR START ADDRESS
>0E19             653  SCBST:   EQU   0E19H           ; SCRIBBLING START ADDRESS

                  655     ; ********************
                  656     ; * POWER UP RESTART *
                  657     ; ********************
                  658           ORG   0
 0000 00          659           NOP                   ; WAIT FOR THINGS TO SETTLE DOW
 0001 F3          660           DI
 0002 AF          661           XOR   A
 0003 D308        662           OUT   (CONCM),A       ; *** SET CONSUMER MODE ***
 0005 C3610C      663           JP    PWRUP

                  665           ORG   8
                  666     ; TRANSFER CONTROL TO RESTART HANDLER
 0008 C30720      667           JP    2007H           ; VECTOR OUT

 000B 1C          669  NUMBAS:  DEFB 1CH
 000C 3C          670           DEFB 3CH
 000D 1C          671           DEFB 1CH
 000E 20          672           DEFB 20H

                  674           ORG   16
 0010 C30A20      675           JP    200AH           ; RESTART 2
 0013 06          676  MENUCL:  DEFB 06H              ; MENU COLORS
 0014 FB          677           DEFB 0FBH
 0015 07          678           DEFB 07H
 0016 52          679           DEFB 52H

                  681           ORG   24
 0018 C30D20      682           JP    200DH           ; RESTART 3
```

```
                684  ; NAME:          PAUSE
                685  ; PURPOSE:       HALT # OF INTERRUPTS
                686  ; INPUT:         B = # OF INTERRUPTS
001B FB         687  MPAUSE: EI
001C 76         688          HALT
001D 10FD       689          DJNZ -1
001F C9         690          RET
```

```
               692         ORG   32
0020 C31020    693         JP    2010H         ; RESTART 4


               695  ; NAME: SET WORD
               696   ; (HL)=DE
0023 73        697  MSETW:  LD    (HL),E
0024 23        698          INC   HL
0025 72        699          LD    (HL),D
0026 C9        700          RET


               702          ORG   40
0028 C31320    703          JP    2013H        ; RESTART 5


002B 210000    705  CONC2.  LD    HL,0         ; ZERO OUT HL
002E C9        706          RET


               708          ORG   48
0030 C31620    709          JP    2016H        ; RESTART 6


0033 00        711  CKSUM1: DEFB 0             ; CHECKSUM


0034 8B01      713  ITAB:   DEFW MACTIN        ; INTERRUPT TRANSFER
0036 01        714          DEFB 1             ; ** SYSTEM REVISION LEVEL


               716          ORG   56
               717   ; NAME:            USER PROGRAM INTERFACE
               718   ; PURPOSE:         TRANSFER OF CONTROL FROM USER TO SYSTEM
               719   ; INPUT:           ROUTINE # FOLLOWS INLINE AFTER RST INSTR
               720   ;                  IF L.O. BIT SET, LOAD ARGUMENTS INLINE F
               721   ; OUTPUT:                 NONE
               722   ; STACK USE:       18 BYTES TOTAL, 16 BYTES ON EXIT
               723   ; SIDE EFFECTS:  REGISTERS AF,BC,DE,HL,IX, AND OLD IY SAV
               724   ; EXPLANATION:
               725   ;  REGISTERS AF,BC,DE,HL,IX, AND PREVIOUS IY ARE PUSHED
               726   ; THE NUMBER FOLLOWING THE RST 56 INSTRUCTION IS USED TO   ;
               727   ; INDEX A JUMP VECTOR GIVING THE STARTING ADDRESS OF THE
               728   ; SYSTEM ROUTINE TO CALL.  IF OPTIONED, INLINE ARGUMENTS
               729   ; ARE COPIED INTO THE CONTEXT AREA.  FOR ARGUMENT ORDERIN
               730   ; SEE INTERPRETER DOCUMENTATION AND APPROP.  TABLES
               731   ; A DUMMY RETURN IS INSERTED WHICH, WHEN RETURNED TO BY
               732   ; SYSTEM ROUTINE, WILL RESTORE THE REGISTER CONTENTS AND
               733   ; RETURN TO THE USER PROGRAM
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND        COMMENT

                734   ;
                735   ;    *** THE UPI HAS BEEN EXTENDED TO SUPPORT USER SUPPLI
                736   ,    ROUTINES.  IF THE CALL INDEX PROVIDED IS NEGATIVE
                737   ;    THEN THE USERS DISPATCH TABLE POINTER (USERTB) IS US
                738   ,    NOTE THAT THE SIGN BIT ISN'T ZAPPED BEFORE BEING
                739   ;    USED AS AN INDEX. THIS MEANS THAT THE USERS DISPATCH
                740   ;    TABLE POINTER SHOULD POINT 128 BYTES BEFORE THE FIRS
0038 E3         741          EX   (SP),HL        ; RETURN ADDRESS TO HL
0039 F5         742          PUSH AF             ; CREATE CONTEXT
003A C5         743          PUSH BC
003B D5         744          PUSH DE
003C DDE5       745          PUSH IX
003E FDE5       746          PUSH IY
0040 FD210000   747          LD   IY,0           ; POINT IY AT CONTEXT
0044 FD39       748          ADD  IY,SP
0046 7E         749          LD   A,(HL)         ; LOAD OPCODE
0047 23         750          INC  HL
0048 117A02     751          LD   DE,RETN        ; DE = RETURN POINT
004B 1F         752          RRA                 ; SUCK WANTED?
004C 3836       753          JR   C,MINTO-$      ; JUMP IF YES
004E E5         754   INTPE: PUSH HL             ; SAVE PC
004F D5         755          PUSH DE             ; SAVE DUMMY RETURN
0050 21CB00     756          LD   HL,SYSDPT
0053 07         757          RLCA
0054 5F         758          LD   E,A
0055 1600       759          LD   D,0
0057 17         760          RLA                 ; USER TABLE WANTED?
0058 3003       761          JR   NC,PUSH1-$
005A 2AFD4F     762          LD   HL,(USERTB)    ; YES - LOAD IT
005D 19         763   PUSH1  ADD  HL,DE
005E 5E         764          LD   E,(HL)
005F 23         765          INC  HL
0060 56         766          LD   D,(HL)
0061 D5         767          PUSH DE
0062 FD660B     768          LD   H,(IY+CBH)
0065 FD6E0A     769          LD   L,(IY+CBL)
0068 FD5603     770   RELD:  LD   D,(IY+CBIXH)
006B FD5E02     771          LD   E,(IY+CBIXL)
006E D5         772          PUSH DE
006F DDE1       773          POP  IX
0071 FD7E09     774          LD   A,(IY+CBA)
0074 FD5605     775   DELOAD: LD  D,(IY+CBD)
0077 FD5E04     776          LD   E,(IY+CBE)
007A C9         777          RET                 ; CALL VIA RETURN
```

```
                 779   ; NAME:            MACRO INTERPRETER
                 780   ; PURPOSE:         INTERPRETING SEQUENCES OF SYSTEM CALLS
                 781   ; INPUT:           ADDRESS OF STRING TO INTERPRET PASSED ON
                 782   ; STACK USE:       NO INCREASE IN DEPTH
                 783   ; EXPLANATION:  IF OPTIONED (BIT 0 OF CALL INDEX SET) THE
                 784   ; ARGUMENT TABLE (MRARGT) IS INDEXED GIVING A MASK WHICH
                 785   ; SPECIFIES HOW TO TRANSFER INLINE ARGUMENTS INTO THE CO
                 786   ; BLOCK.   THIS MASK IS FORMATED AS FOLLOWS:
                 787   ;
                 788   ;
                 789   ; *********************************
                 790   ; * 7 * 6 * 5 * 4 * 3 * 2 * 1 * 0 *
                 791   ; *********************************
                 792   ; * H * L * A * IX* B * C * D * E *
                 793   ; *********************************
                 794   ; ARGUMENTS MUST FOLLOW THE CALL INDEX IN THE FOLLOWING
                 795   ; (OMITING UNUSED ARGUMENTS, OF COURSE)
                 796   ; (INDEX),IXL,IXH,E,D,C,B,A,L,H
                 797   ;
                 798   ;        THE SIMULATED PC IS SAVED AND A DUMMY RETURN IS
                 799   ; INSERTED ON THE STACK.   THE UPI DISPATCHING ROUTINE IS
                 800   ; THEN ENTERED AT 'INTPE', WHICH EFFECTS A CONTROL TRANS
                 801   ; TO THE CALLED ROUTINE.   WHEN THE CALLED ROUTINE RETURN
                 802   ; IT WILL COME BACK HERE TO INTERPRET THE NEXT MACRO INS
                 803   ; NOTE THAT THIS ROUTINE IS REENTRANT, THEREFORE THE CAL
                 804   ; ROUTINE MAY RECUR BACK THRU HERE, IF IT FEELS LIKE IT.
                 805   ; ** THE UPI HAS BEEN EXTENDED TO SUPPORT USER PROVIDED
                 806   ; SYSTEM ROUTINES.  IF A NEGATIVE CALL INDEX IS ENCOUNTER
                 807   ; BY THE INTERPRETER, AND 'SUCK INLINE' IS OPTIONED, THE
                 808   ; USER MACRO ROUTINE ARGUMENT TABLE IS INDEXED FOR A
                 809   ; PARAMETER MASK.  THE ADDRESS OF THIS TABLE IS ASSUMED
                 810   ; TO BE IN (UMARGT),(UMARGT+1).   THIS POINTER SHOULD
                 811   ; POINT 64 BYTES BEFORE THE FIRST REAL ENTRY.
                 812   ; I.E. LD       HL,USERMT-64     ; WHERE USERMT POINTS AT
                 813   ;      LD       (UMARGT),HL
007B D1          814  MINTPC: POP   DE              ; DISCARD DUMMY RETURN FROM UPI
007C             815  RENTER:
007C E1          816          POP   HL              ; POP OFF PC


                 818  ; NAME:           MCALL
                 819  ; PURPOSE:        CALL INTERPRETER SUBROUTINE
                 820  ; INPUT:          HL = ROUTINE ADDRESS
                 821  ; NOTES:          ROUTINE MAY BE CALLED FROM MACHINE LANGUA
                 822  ;                 ANOTHER INTERPRETED SEQUENCE
                 823  ;                 STACK DEPTH INCREASED BY 4 BY CALL
007D 7E          824  MMCALL: LD    A,(HL)          ; GET OPCODE
007E 23          825          INC   HL
007F CB3F        826          SRL   A
0081 117C00      827          LD    DE,RENTER       ; LOAD INTERPRETER DUMMY RETURN
0084 D5          828  MINTO:  PUSH  DE              ; SAVE DUMMY RETURN
0085 4F          829          LD    C,A             ; INDEX TO C
0086 3012        830          JR    NC,MINT2-$      ; JUMP IF NO LOAD WANTED
0088 EB          831          EX    DE,HL
0089 0600        832          LD    B,0
```

```
008B 214B01    833          LD   HL,MRARGT     ; LOAD SYSTEM ARG TABLE
008E CB77      834          BIT  6,A           ; USE USER TABLE?
0090 2803      835          JR   Z,MINT1-$     ; JUMP IF NO
0092 2AFB4F    836          LD   HL,(UMARGT)
0095 09        837  MINT1:  ADD  HL,BC         ; INDEX TABLE
0096 46        838          LD   B,(HL)
0097 CDA800    839          CALL MSUCK1        ; CALL SUCK ROUTINE
009A D1        840  MINT2:  POP  DE            ; DUMMY RETURN TO DE, HL = PC
009B 79        841          LD   A,C           ; GET CALL INDEX BACK
009C FD4607    842          LD   B,(IY+CBB)    ; RESTORE CLOBBERED REGISTERS
009F FD4E06    843          LD   C,(IY+CBC)
00A2 18AA      844          JR   INTPE-$       ; JOIN NORMAL UPI DISPATCH SEQU

               846   ; NAME:            SUCK INLINE ARGUMENTS
               847   ; PURPOSE:         TRANSFER OF INLINE ARGS INTO CONTEXT BLO
               848   ; INPUT:           B = ARG LOAD MASK (SEE INTERPRETER COMME
               849   ; OUTPUT:          HL = UPDATED PC
               850   ; EXPLANATION: THIS ROUTINE IMPLEMENTS A MACRO LOAD INST
               851   ; IT IS USED BY THE INTERPRETER AS WELL.  A ONE BIT IN T
               852   ; INLINE LOAD MASK MEANS TRANSFER THE NEXT INLINE BYTE I
               853   ; A ZERO BIT MEANS 'ADVANCE CONTEXT BLOCK POINTER'
               854   ; TWO ENTRY POINTS ARE DEFINED, ONE FOR THE SUCK MACRO I
               855   ; THE OTHER FOR THE INTERPRETER TO USE
               856   ; SUCK MACRO ENTRY:
00A4 E1        857  MSUCK:  POP  HL            ; RETURN ADDRESS TO HL
00A5 D1        858          POP  DE            ; POP OFF PC
               859    ; *** BYTE SAVING TRICK *** REPLACE WITH LD HL,REENTRY
00A6 23        860          INC  HL            ; ADVANCE TO REENTRY (MINTO)
00A7 E5        861          PUSH HL
               862    ; FALL INTO ...
00A8 CB60      863  MSUCK1: BIT  4,B           ; IX LOAD WANTED?
00AA 280A      864          JR   Z,MSUCK2-$    ; MSUCK2 IF NOT
00AC 1A        865          LD   A,(DE)
00AD 13        866          INC  DE
00AE FD7702    867          LD   (IY+CBIXL),A
00B1 1A        868          LD   A,(DE)
00B2 13        869          INC  DE
00B3 FD7703    870          LD   (IY+CBIXH),A
00B6 FDE5      871  MSUCK2: PUSH IY            ; LET HL = IY
00B8 E1        872          POP  HL
00B9 23        873          INC  HL            ; + 4
00BA 23        874          INC  HL
00BB 23        875          INC  HL
00BC 23        876          INC  HL
00BD CBA0      877          RES  4,B           ; KILL IX BIT
               878    ;  SUCK IN LOOP
00BF CB38      879  MSUCK3: SRL  B
00C1 3003      880          JR   NC,MSUCK5-$   ; MSUCK5 IF NOT THIS TIME
00C3 1A        881          LD   A,(DE)        ; GET INLINE BYTE
00C4 13        882          INC  DE
00C5 77        883          LD   (HL),A        ; STUFF INTO CB
00C6 23        884  MSUCK5: INC  HL            ; BUMP CB POINTER
               885    ; ** THIS CODE ASSUMES THAT STATUS OF 'SRL' IS PRESERVE
00C7 20F6      886          JR   NZ,MSUCK3-$   ; JUMP BACK IF MORE TO DO
00C9 EB        887          EX   DE,HL         ; HL = PC
00CA C9        888          RET                ; THEN QUIT
```

```
                890    ; ****************************
                891    ; * UPI ROUTINE ADDRESS TABLE *
                892    ; ****************************
00CB 7B00       893  SYSDPT:  DEFW MINTPC
00CD 7902       894           DEFW MXINTC
00CF 3206       895           DEFW MRCALL
00D1 7D00       896           DEFW MMCALL
00D3 730B       897           DEFW MMRET
00D5 C40A       898           DEFW MMJUMP
00D7 A400       899           DEFW MSUCK
00D9 8B01       900           DEFW MACTIN
00DB 7E04       901           DEFW TIMEY
00DD 0805       902           DEFW MUZSET
00DF FC05       903           DEFW MUZSTP
00E1 CF03       904           DEFW MSETUP
00E3 DB01       905           DEFW MCOLOR
00E5 EE0A       906           DEFW MFILL
00E7 B206       907           DEFW MPAINT
00E9 FE06       908           DEFW MVWRIT
00EB 0B07       909           DEFW MWRITR
00ED 1507       910           DEFW MWRITP
00EF 1907       911           DEFW MWRIT
00F1 1C07       912           DEFW MWRITA
00F3 7D07       913           DEFW MVBLAN
00F5 9E07       914           DEFW MBLANK
00F7 B903       915           DEFW MSAVE
00F9 AD07       916           DEFW MREST
00FB 6A02       917           DEFW MSCROL
00FD E107       918           DEFW DISPCH
00FF C407       919           DEFW STRNEW
0101 EB0B       920           DEFW BCDISP
0103 F60A       921           DEFW MRELAB
0105 FB0A       922           DEFW MRELA1        ; RELAB1
0107 5606       923           DEFW MVECTC
0109 3306       924           DEFW MVECT
010B C90A       925           DEFW MKCTAS
010D AC01       926           DEFW MENTRY        ; SENTRY
010F 0C06       927           DEFW MDOIT         ; DOIT
0111 0B06       928           DEFW MDOITB
0113 BA01       929           DEFW MPIZBK        ; PIZBRK
0115 970C       930           DEFW MMENU
0117 FB0C       931           DEFW MGETP
0119 310D       932           DEFW MGETN
011B 1B00       933           DEFW MPAUSE        ; PAUSE
011D CC0B       934           DEFW MDISTI        ; DISPLAY TIME
011F 150C       935           DEFW MINCSC        ; INC SCORE
0121 760B       936           DEFW INXNIB        ; INDEXN
0123 900B       937           DEFW PUTNIB        ; STOREN
0125 AC0B       938           DEFW MINDW         ; INDEXW
0127 BD0B       939           DEFW MINDB         ; INDEXB
0129 4B0B       940           DEFW MMOVE         ; MOVE
012B AA0D       941           DEFW MSHFTU
012D 2103       942           DEFW BCDAD
012F 1F03       943           DEFW BCDSB
0131 DE02       944           DEFW BCDML
0133 8402       945           DEFW BCDDV
```

```
0135 6403      946         DEFW BCDCS
0137 4103      947         DEFW BCDNG
0139 6E03      948         DEFW SDADD
013B 2903      949         DEFW SDSMG
013D 5603      950         DEFW SDABS
013F 4C03      951         DEFW SNEGT
0141 7F03      952         DEFW MRANGE
0143 410C      953         DEFW MQUIT
0145 6C03      954         DEFW MSETB
0147 2300      955         DEFW MSETW
0149 4002      956         DEFW MMTD



               958   ; MACRO ROUTINES ARGUMENT MASK TABLE
               959   ; FORMAT:
               960   ; *********************************
               961   ; * 7 * 6 * 5 * 4 * 3 * 2 * 1 * 0 *
               962   ; *********************************
               963   ; * H * L * A * IX* B * C * D * E *
               964   ; *********************************
               965   ; ARGUMENTS MUST FOLLOW THE CALL INDEX IN THE FOLLOWING
               966   ; (OMITING UNUSED ARGUMENTS, OF COURSE)
               967   ; (INDEX),IXL,IXH,E,D,C,B,A,L,H
014B 00        968  MRARGT: DEFB 0            ; INTPC
014C 00        969         DEFB 0            ; XINTC
014D C0        970         DEFB 11000000B    ; RCALL
014E C0        971         DEFB 11000000B    ; MCALL
014F 00        972         DEFB 0            ; MRET
0150 C0        973         DEFB 11000000B    ; MJUMP
0151 08        974         DEFB 00001000B    ; SUCK
0152 00        975         DEFB 0            ; ACTINT
0153 04        976         DEFB 00000100B    ; DECCTS
0154 F0        977         DEFB 11110000B    ; BMUSIC
0155 00        978         DEFB 0            ; EMUSIC
0156 2A        979         DEFB 00101010B    ; SETOUT
0157 C0        980         DEFB 11000000B    ; COLSET
0158 2F        981         DEFB 00101111B    ; FILL
0159 2F        982         DEFB 00101111B    ; RECTAN
015A D0        983         DEFB 11010000B    ; VWRITR
015B E3        984         DEFB 11100011B    ; WRITR
015C E3        985         DEFB 11100011B    ; WRITP
015D EF        986         DEFB 11101111B    ; WRIT
015E EF        987         DEFB 11101111B    ; WRITA
015F 13        988         DEFB 00010011B    ; VBLANK
0160 CB        989         DEFB 11001011B    ; BLANK
0161 CF        990         DEFB 11001111B    ; SAVE
0162 C3        991         DEFB 11000011B    ; RESTORE
0163 CF        992         DEFB 11001111B    ; SCROLL
0164 27        993         DEFB 00100111B    ; NEW DISCHR
0165 C7        994         DEFB 11000111B    ; NEW DISSTR
0166 CF        995         DEFB 11001111B    ; DISNUM
0167 20        996         DEFB 00100000B    ; RELABS
0168 20        997         DEFB 00100000B    ; RELAB1
0169 D4        998         DEFB 11010100B    ; VECTC
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND      COMMENT

016A D0         999          DEFB 11010000B    ; VECT
016B 00        1000          DEFB 0            ; KCTASC
016C 03        1001          DEFB 00000011B    ; SENTRY
016D C0        1002          DEFB 11000000B    ; DOIT
016E C0        1003          DEFB 11000000B    ; DOITB
016F 00        1004          DEFB 0            ; PIZBRK
0170 C3        1005          DEFB 11000011B    ; MENU
0171 EC        1006          DEFB 11101100B    ; GET PARAMETER
0172 CF        1007          DEFB 11001111B    ; GET NUMBER
0173 08        1008          DEFB 00001000B    ; PAUSE
0174 07        1009          DEFB 00000111B    ; DISTIM
0175 C0        1010          DEFB 11000000B    ; INCSCR
0176 C0        1011          DEFB 11000000B    ; INDEXN
0177 C0        1012          DEFB 11000000B    ; STOREN
0178 C0        1013          DEFB 11000000B    ; INDEXW
0179 C0        1014          DEFB 11000000B    ; INDEXB
017A CF        1015          DEFB 11001111B    ; MOVE
017B C8        1016          DEFB 11001000B    ; SHIFTU
017C CB        1017          DEFB 11001011B    ; BCDADD
017D CB        1018          DEFB 11001011B    ; BCDSUB
017E CB        1019          DEFB 11001011B    ; BCDMUL
017F CB        1020          DEFB 11001011B    ; BCDDIV
0180 C8        1021          DEFB 11001000B    ; BCDCHS
0181 0B        1022          DEFB 00001011B    ; BCDNEG
0182 CB        1023          DEFB 11001011B    ; DADD
0183 0B        1024          DEFB 00001011B    ; DSMG
0184 0B        1025          DEFB 00001011B    ; DABS
0185 C8        1026          DEFB 11001000B    ; NEGT
0186 20        1027          DEFB 00100000B    ; RANGED
0187 00        1028          DEFB 00000000B    ; QUIT
0188 E0        1029          DEFB 11100000B    ; SET BYTE
0189 C3        1030          DEFB 11000011B    ; SET WORD
018A C7        1031          DEFB 11000111B    ; MASK TO DELTAS



               1033   ; DOES 4 60TH SEC COUNTERS IN CT0-3
018B F3        1034  MACTIN: DI                ; MAKE SURE INTERRUPT IS DISABL
018C F5        1035          PUSH AF
018D C5        1036          PUSH BC
018E D5        1037          PUSH DE
018F E5        1038          PUSH HL
0190 ED5E      1039          IM   2
0192 3E00      1040          LD   A,ITAB.SHR.8
0194 ED47      1041          LD   I,A
0196 3EC8      1042          LD   A,200
0198 D30F      1043          OUT  (INLIN),A
019A 3E34      1044          LD   A,ITAB&0FFH
019C D30D      1045          OUT  (INFBK),A
019E CDA004    1046          CALL TIMEZ        ; UPDATE TIMOUT,MUSIC AND SECON
01A1 0E0F      1047          LD   C,0FH        ; USE CT0-3
01A3 CD7E04    1048          CALL TIMEY        ; DEC CT0-3
01A6 E1        1049          POP  HL
01A7 D1        1050          POP  DE
01A8 C1        1051          POP  BC
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

01A9 F1        1052           POP  AF
01AA FB        1053           EI
01AB C9        1054           RET



               1056  ; ROUTINE:  SENTRY
               1057  ; PURPOSE:  TO WAIT FOR CHANGE OF PROGRAM STATUS
               1058  ; IN EITHER THE PORTS OR THE TIMER-COUNTERS.
               1059  ; IN ADDITION IT CHECKS TIMOUT FOR LONG PERIODS OF IN-
               1060  ; ACTIVITY.
               1061  ; ** IS VECTOR OUT FLAG SET??
01AC 3AFA4F    1062  MENTRY:  LD   A,(SENFLG)
01AF FEAA      1063           CP   0AAH
01B1 CA1920    1064           JP   Z,2019H        ; YES - JUMP OUT
01B4 3AEC4F    1065           LD   A,(TIMOUT)     ; CHECK IF TIME TO BLAKOUT
01B7 B7        1066           OR   A
01B8 202B      1067           JR   NZ,TTEST-$
01BA AF        1068  MPIZBK:  XOR  A              ; TIME TO SHUT DOWN
01BB F3        1069           DI
01BC D315      1070           OUT  (VOLC),A       ; TURN OFF SOUNDS
01BE D316      1071           OUT  (VOLAB),A
01C0 010B08    1072           LD   BC,COLBX+8*256
01C3 ED79      1073           OUT  (C),A          ; PAINT IT BLACK
01C5 10FC      1074           DJNZ -2
01C7 111402    1075  PBLP:    LD   DE,AKEYS
01CA CDF40C    1076           CALL FINDL3         ; CALL STORE DE INTO CONTEXT RO
01CD CDE501    1077           CALL TTEST          ; WAIT FOR SOMETHING TO HAPPEN
01D0 3C        1078           INC  A
01D1 20E7      1079           JR   NZ,MPIZBK-$
01D3 FD360900  1080           LD   (IY+CBA),0
01D7 FB        1081           EI
01D8 2AE84F    1082           LD   HL,(COLLST)   ; GET SAVED COLORS
01DB 22E84F    1083  MCOLOR:  LD   (COLLST),HL   ; SAVE COLORS FOR FUTURE
01DE 010B08    1084           LD   BC,800H+COLBX
01E1 EDB3      1085           OTIR                ; RESET THE COLORS
01E3 AF        1086           XOR  A
01E4 C9        1087           RET
01E5 CDEC03    1088  TTEST    CALL TRCHK
01E8 FD7709    1089           LD   (IY+CBA),A
01EB FD7007    1090           LD   (IY+CBB),B
01EE FE13      1091           CP   SKYD
01F0 D8        1092           RET  C
01F1 FE1C      1093           CP   POTO
01F3 D0        1094           RET  NC
01F4 3EFF      1095           LD   A,0FFH
01F6 32EC4F    1096           LD   (TIMOUT),A
01F9 C9        1097           RET
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND         COMMENT

01FA C40D      1099  CALCL:   DEFW SCBL
01FC DD0D      1100           DEFW PNCALC
01FE 2010      1101           DEFW CALCST          ; START OF CALCULATOR



               1103    ; SYSTEM ROUTINES JUMP VECTOR
               1104           ORG  200H
0200 C3A004    1105           JP   TIMEZ           ; DO TIMER & MUSIC
0203 C37B04    1106           JP   TIMEX           ; DECTMR

0206 20        1108  SYSFNT:  DEFB 20H
0207 08        1109           DEFB 8
0208 08        1110           DEFB 8
0209 01        1111           DEFB 1
020A 07        1112           DEFB 7
020B E408      1113           DEFW LRGCHR

020D A0        1115  SMLFNT:  DEFB 0A0H
020E 04        1116           DEFB 4
020F 06        1117           DEFB 6
0210 01        1118           DEFB 1
0211 05        1119           DEFB 5
0212 BF0A      1120           DEFW SMLCHR

               1122  ; ALLKEYS MASK
0214 3F        1123  AKEYS    DEFB 3FH
0215 3F        1124           DEFB 3FH
0216 3F        1125           DEFB 3FH
0217 3F        1126           DEFB 3FH

               1128  ; HEAD OF ONBOARD MENU
0218 BE0D      1129  GUNLNK:  DEFW CML
021A CA0D      1130           DEFW PNGF
021C DE17      1131           DEFW GFSTRT
021E 4D415820  1132           DEFM 'MAX SCORE'
0227 00        1133           DEFB 0
0228 23204F46  1134           DEFM '# OF PLAYERS'
0234 00        1135           DEFB 0
0235 23204F46  1136           DEFM '# OF GAMES'
023F 00        1137           DEFB 0
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

              1139  , NAME:          CONVERT MASK TO DELTAS
              1140  ; INPUT:         B = JOYSTICK MASK
              1141  ;                C = FLOP STATUS (MR FLOP BIT SET IF FLOP
              1142  ;                DE = X POSITIVE DELTA
              1143  ;                HL = Y POSITIVE DELTA
0240 CD5602   1144  MMTD:   CALL CONCPL         ; HANDLE Y
0243 EB       1145          EX   DE,HL
0244 CB71     1146          BIT  MRFLOP,C       ; FLOP SET?
0246 2807     1147          JR   Z,MMTD2-$      ; YES - DOIT
0248 78       1148          LD   A,B            ; NO - GET MASK
0249 E603     1149          AND  3
024B 2801     1150          JR   Z,MMTD1-$
024D 2F       1151          CPL                 ; INVERT IF NOT ZERO
024E 47       1152  MMTD1:  LD   B,A
024F CD5602   1153  MMTD2:  CALL CONCPL         ; PROCESS X
0252 EB       1154          EX   DE,HL
0253 C3B80B   1155          JP   STHLDE         ; STORE HL,DE AND QUIT


              1157  ; SUBROUTINE TO CONDITIONALLY COMPLEMENT OR ZERO HL
0256 CB08     1158  CONCPL: RRC  B
0258 300A     1159          JR   NC,CONC1-$     ; JUMP IF NOT UP
025A 7D       1160          LD   A,L
025B 2F       1161          CPL
025C 6F       1162          LD   L,A
025D 7C       1163          LD   A,H
025E 2F       1164          CPL
025F 67       1165          LD   H,A
0260 23       1166          INC  HL
0261 CB08     1167          RRC  B
0263 C9       1168          RET
0264 CB08     1169  CONC1:  RRC  B              ; DOWN SET?
0266 D8       1170          RET  C              ; QUIT IF SO
0267 C32B00   1171          JP   CONC2          ; JUMP TO ZERO OUT


              1173  ; NAME:          SCROLL MEMORY BLOCK
              1174  ; INPUT:         B = NUMBER OF LINES TO SCROLL
              1175  ;                C = NUMBER OF BYTES ON LINE TO SCROLL
              1176  ;                DE = LINE INCREMENT
              1177  ;                HL = FIRST LINE TO SCROLL
026A AF       1178  MSCROL: XOR  A
026B C5       1179  MSCRL1: PUSH BC            ; SAVE COUNTERS
026C D5       1180          PUSH DE
026D 47       1181          LD   B,A
026E EB       1182          EX   DE,HL
026F 19       1183          ADD  HL,DE          ; ADD INCREMENT TO LINE
0270 E5       1184          PUSH HL
0271 EDB0     1185          LDIR                ;
0273 E1       1186          POP  HL
0274 D1       1187          POP  DE
0275 C1       1188          POP  BC
0276 10F3     1189          DJNZ MSCRL1-$
0278 C9       1190          RET
```

```
               1192   ;  NAME:            MACRO INTERPRETER EXIT WITH CONTEXT REST
               1193   ;  PURPOSE:         QUIT INTERPRETING AND GO HOME
0279 E1        1194  MXINTC:  POP   HL                ;  THROW OUT DUMMY RETURN
               1195   ;  NAME:            RETURN FROM SYSTEM CALL
               1196   ;  PURPOSE:         RETURNING TO USER AND RESTORATION OF REG
027A E1        1197  RETN:    POP   HL                ;  RETURN ADDRESS TO HL
027B FDE1      1198           POP   IY
027D DDE1      1199           POP   IX
027F D1        1200           POP   DE
0280 C1        1201           POP   BC
0281 F1        1202           POP   AF
0282 E3        1203           EX    (SP),HL           ;  STK=RETURN, HL=OLD HL
0283 C9        1204           RET


               1206   ;  NAME:          BCD DIVIDE
               1207   ;
0284 CDC002    1208  BCDDV:   CALL  GNACC             ;  GENERATE ACCUMULATOR
0287 E3        1209           EX    (SP),HL           ;  HL = ACC, TOP = ARG2
0288 C5        1210           PUSH  BC
0289 0600      1211           LD    B,0
028B 79        1212           LD    A,C
028C CB39      1213           SRL   C
028E 09        1214           ADD   HL,BC
028F 4F        1215           LD    C,A
0290 EB        1216           EX    DE,HL             ;  HL = ARG1, DE = ACC
0291 EDB0      1217           LDIR                    ;  HL = ARG1 FLAG+1
0293 C1        1218           POP   BC
0294 D1        1219           POP   DE
0295 2B        1220           DEC   HL                ;
0296 E3        1221           EX    (SP),HL           ;  HL = ARG2, TOP = ARG1 FLAG
0297 C5        1222           PUSH  BC
0298 0600      1223           LD    B,0
029A 09        1224           ADD   HL,BC             ;  HL = ACC+SIZE/2
029B C1        1225           POP   BC
029C 0D        1226           DEC   C                 ;  DECREMENT SIZE
029D EB        1227           EX    DE,HL             ;  HL = ARG2, DE = ACC, TOP = AR
029E 1B        1228           DEC   DE                ;
029F 1B        1229  DIV1:    DEC   DE
02A0 AF        1230           XOR   A
02A1           1231           SYSTEM NEGT             ;  ARG2 = -ARG2 (10S COMP)
02A1 FF        1231 +         RST   56
02A2 74        1231 +         DEFB  NEGT
               1231 +         IF    NEGT.EQ.INTPC
               1231 +         ENDIF
02A3           1232  DIV2:    SYSTEM DADD             ;  SUBTRACT UNTIL BORROW
02A3 FF        1232 +         RST   56
02A4 6E        1232 +         DEFB  DADD
               1232 +         IF    DADD.EQ.INTPC
               1232 +         ENDIF
02A5 380A      1233           JR    C,DIV3-$
02A7 3C        1234           INC   A                 ;  OR UNTIL LOOP COUNT > 99
02A8 27        1235           DAA
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

02A9 20F8     1236        JR   NZ,DIV2-$
02AB E1       1237        POP  HL
02AC 36FF     1238        LD   (HL),0FFH
02AE C1       1239        POP  BC
02AF 186A     1240        JR   MULT6-$
02B1          1241  DIV3:   SYSTEM NEGT
02B1 FF       1241 +        RST  56
02B2 74       1241 +        DEFB NEGT
              1241 +        IF   NEGT.EQ.INTPC
              1241 +        ENDIF
02B3          1242        SYSTEM DADD
02B3 FF       1242 +        RST  56
02B4 6E       1242 +        DEFB DADD
              1242 +        IF   DADD.EQ.INTPC
              1242 +        ENDIF
02B5 E3       1243        EX   (SP),HL      ; HL = ARG1
02B6 2B       1244        DEC  HL
02B7 77       1245        LD   (HL),A       ; SAVE ANSWER IN ARG1
02B8 E3       1246        EX   (SP),HL
02B9 0D       1247        DEC  C
02BA 20E3     1248        JR   NZ,DIV1-$
02BC E1       1249        POP  HL
02BD C1       1250        POP  BC
02BE 1855     1251        JR   DIV4-$
              1252  ; SUBROUTINE TO GENERATE ACCUMULATOR ON THE STACK
02C0 DDE1     1253  GNACC:  POP  IX
02C2 AF       1254        XOR  A
02C3 4F       1255        LD   C,A
02C4          1256        SYSTEM DABS       ;ARG1=ABS VALUE
02C4 FF       1256 +        RST  56
02C5 72       1256 +        DEFB DABS
              1256 +        IF   DABS.EQ.INTPC
              1256 +        ENDIF
02C6 EB       1257        EX   DE,HL
02C7          1258        SYSTEM DABS       ;ARG2=ABS VALUE
02C7 FF       1258 +        RST  56
02C8 72       1258 +        DEFB DABS
              1258 +        IF   DABS.EQ.INTPC
              1258 +        ENDIF
02C9 EB       1259        EX   DE,HL        ;FLAG=1 IF NEG ANS, ELSE POS
02CA 67       1260        LD   H,A
02CB 6F       1261        LD   L,A
02CC 78       1262        LD   A,B
02CD E5       1263  MULT1   PUSH HL           ;GENERATE ACC ON STACK
02CE 10FD     1264        DJNZ MULT1-$
02D0 47       1265        LD   B,A          ;RESTORE SIZE
02D1 39       1266        ADD  HL,SP
02D2 C5       1267        PUSH BC           ; SAVE SIGN
02D3 E5       1268        PUSH HL           ;SAVE STACK POINTER
02D4 E5       1269        PUSH HL           ;SAVE ACC POINTER
02D5 FD660B   1270        LD   H,(IY+CBH)   ;RESTORE ARG2 POINTER
02D8 FD6E0A   1271        LD   L,(IY+CBL)
02DB 48       1272        LD   C,B
02DC DDE9     1273        JP   (IX)
              1274        ;DECIMAL MULTIPLY
              1275        ;GIVEN:    DE>ARG1, HL>ARG2, B=SIZE/2
              1276        ;          (SIZE/2-1 ASSUMED EVEN)
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               1277           ;RETURNED:  ARG1=ANSWER, C>0 ON OVERFLOW
               1278           ;
               1279           ;
02DE CDC002    1280  BCDML:   CALL GNACC          ;  GENERATE ACCUM
02E1 7E        1281  MULT2    LD   A,(HL)         ;A=MULT LOOP COUNT
02E2 23        1282           INC  HL
02E3 E3        1283           EX   (SP),HL        ;HL>DEC ACC
02E4 A7        1284           AND  A              ;IF A=0, SKIP MULT LOOP
02E5 2809      1285           JR   Z,MULT4-$
02E7 EB        1286           EX   DE,HL
02E8           1287  MULT3:   SYSTEM DADD         ;  ELSE MULTIPLY
02E8 FF        1287 +         RST  56
02E9 6E        1287 +         DEFB DADD
               1287 +         IF   DADD.EQ.INTPC
               1287 +         ENDIF
02EA A7        1288           AND  A              ;  CLEAR THE CARRY BIT
02EB 3D        1289           DEC  A              ;DECIMAL DECREMENT
02EC 27        1290           DAA
02ED 20F9      1291           JR   NZ,MULT3-$
02EF EB        1292           EX   DE,HL
02F0 23        1293  MULT4:   INC  HL             ;INCREMENT DECIMAL ACC
02F1 E3        1294           EX   (SP),HL        ;HL>ARG2
02F2 0D        1295           DEC  C
02F3 20EC      1296           JR   NZ,MULT2-$
02F5 E1        1297           POP  HL
02F6 E1        1298           POP  HL             ;RESTORE STACK POINTER
02F7 C1        1299           POP  BC             ;  RESTORE SIGN
02F8 D5        1300           PUSH DE
02F9 C5        1301           PUSH BC
02FA 48        1302           LD   C,B
02FB 0600      1303           LD   B,0
02FD CB39      1304           SRL  C
02FF 09        1305           ADD  HL,BC
0300 CB21      1306           SLA  C
0302 EDB0      1307           LDIR
0304 C1        1308           POP  BC
0305 C5        1309           PUSH BC             ;CHECK FOR OVERFLOW
0306 CB38      1310           SRL  B
0308 AF        1311           XOR  A
0309 B6        1312  MULT5:   OR   (HL)
030A 23        1313           INC  HL
030B 10FC      1314           DJNZ MULT5-$
030D A7        1315           AND  A              ;  SET FLAGS
030E 2803      1316           JR   Z,MULT7-$
0310 3EFF      1317           LD   A,0FFH
0312 12        1318           LD   (DE),A
0313 C1        1319  MULT7:   POP  BC             ;CHECK SIGN AND
0314 E1        1320           POP  HL
0315 CB41      1321  DIV4:    BIT  0,C            ;NEGATE ARG1 IF NECESSARY
0317 2802      1322           JR   Z,MULT6-$
0319           1323           SYSTEM BCDCHS
0319 FF        1323 +         RST  56
031A 6A        1323 +         DEFB BCDCHS
               1323 +         IF   BCDCHS.EQ.INTPC
               1323 +         ENDIF
031B E1        1324  MULT6:   POP  HL             ;RESTORE ORIGINAL STACK POINTER
031C 10FD      1325           DJNZ MULT6-$
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

031E C9        1326          RET
               1327          ;BCD SUBTRACT & ADD
               1328          ;
               1329          ;GIVEN:    DE>ARG1, HL>ARG2
               1330          ;          B=SIZE/2+1
               1331          ;RETURNED: ARG1=ANSWER
031F           1332  BCDSB:  SYSTEM BCDCHS
031F FF        1332 +        RST  56
0320 6A        1332 +        DEFB BCDCHS
               1332 +        IF   BCDCHS.EQ.INTPC
               1332 +        ENDIF
0321           1333  BCDAD:  SYSTEM BCDNEG
0321 FF        1333 +        RST  56
0322 6C        1333 +        DEFB BCDNEG
               1333 +        IF   BCDNEG.EQ.INTPC
               1333 +        ENDIF
0323 EB        1334          EX   DE,HL
0324           1335          SYSTEM BCDNEG
0324 FF        1335 +        RST  56
0325 6C        1335 +        DEFB BCDNEG
               1335 +        IF   BCDNEG.EQ.INTPC
               1335 +        ENDIF
0326 EB        1336          EX   DE,HL
0327           1337          SYSTEM DADD
0327 FF        1337 +        RST  56
0328 6E        1337 +        DEFB DADD
               1337 +        IF   DADD.EQ.INTPC
               1337 +        ENDIF
               1338  ; AND FALL INTO
               1339          ;
               1340          ;
               1341          ;DECIMAL SIGNED MAGNITUDE
               1342          ;
               1343          ;GIVEN:    DE>ARG (10'S COMPLEMENT)
               1344          ;          B=SIZE/2+1
               1345          ;RETURNED: ARG (SIGNED MAGNITUDE)
               1346          ;
0329 68        1347  SDSMG:  LD   L,B          ;HL>ARG+B-1 (SIGN BYTE)
032A 2D        1348          DEC  L
032B 2600      1349          LD   H,0
032D 19        1350          ADD  HL,DE
032E 7E        1351          LD   A,(HL)       ;IF POS (SIGN NIBBLE<5)
032F FE50      1352          CP   50H
0331 D8        1353          RET  C            ;EXIT
0332 EB        1354          EX   DE,HL
0333 3E00      1355  SDSMG1: LD   A,0          ;ELSE 10'S COMPLEMENT
0335 9E        1356          SBC  A,(HL)
0336 27        1357          DAA
0337 77        1358          LD   (HL),A
0338 23        1359          INC  HL
0339 10F8      1360          DJNZ SDSMG1-$
033B 2B        1361          DEC  HL           ;AND SET SIGN BIT
033C 7E        1362          LD   A,(HL)
033D F680      1363          OR   80H
033F 77        1364          LD   (HL),A
0340 C9        1365          RET
               1366          ;
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               1367           ;
               1368           ;BCD NEGATE
               1369           ;
               1370           ;GIVEN:     DE>ARG (SIGNED MAGNITUDE)
               1371           ;           B=SIZE/2+1
               1372           ;RETURNED:  ARG (10'S COMPLEMENT)
               1373           ;
0341 68        1374  BCDNG:   LD   L,B           ;HL>ARG+B-1 (SIGN BYTE)
0342 2D        1375           DEC  L
0343 2600      1376           LD   H,0
0345 19        1377           ADD  HL,DE
0346 CB7E      1378           BIT  7,(HL)        ;EXIT IF POS
0348 C8        1379           RET  Z
0349 3600      1380           LD   (HL),0        ; CLEAR SIGN BYTE
034B EB        1381           EX   DE,HL
034C AF        1382  SNEGT:   XOR  A             ; CLEAR CARRY
034D 3E00      1383  BCDNG1:  LD   A,0           ;ELSE 10'S COMPLEMENT
034F 9E        1384           SBC  A,(HL)
0350 27        1385           DAA
0351 77        1386           LD   (HL),A
0352 23        1387           INC  HL
0353 10F8      1388           DJNZ BCDNG1-$
0355 C9        1389           RET
               1390           ;
               1391           ;
               1392           ;DECIMAL ABSOLUTE
               1393           ;
               1394           ;GIVEN:     DE>ARG (SIGNED MAGNITUDE)
               1395           ;           B=SIZE/2+1
               1396           ;RETURNED:  C=C+1 IF SIGN BIT CLEARED
               1397           ;
0356 68        1398  SDABS:   LD   L,B
0357 2600      1399           LD   H,0
0359 2D        1400           DEC  L
035A 19        1401           ADD  HL,DE
035B CB7E      1402           BIT  7,(HL)
035D C8        1403           RET  Z
035E 3600      1404           LD   (HL),0
0360 FD3406    1405           INC  (IY+CBC)
0363 C9        1406           RET
               1407           ;
               1408           ;
               1409           ;BCD CHANGE SIGN
               1410           ;
               1411           ;GIVEN:     HL>ARG  B=SIZE/2+1
               1412           ;           (SIGNED MAGNITUDE)
               1413           ;RETURNED:  ARG SIGN BIT COMPLEMENTED
               1414           ;
0364 48        1415  BCDCS:   LD   C,B
0365 0600      1416           LD   B,0
0367 0D        1417           DEC  C
0368 09        1418           ADD  HL,BC
0369 7E        1419           LD   A,(HL)
036A EE80      1420           XOR  80H
               1421  ; NAME:         SET BYTE
036C 77        1422  MSETB:   LD   (HL),A
036D C9        1423           RET
```

```
                    1424            ;
                    1425            ;
                    1426            ;DECIMAL ADD
                    1427            ;
                    1428            ;GIVEN:     DE>ARG1  HL>ARG2 (10'S COMPLEMENT)
                    1429            ;           B=SIZE/2+1
                    1430            ;RETURNED:  ARG1=ANSWER (10'S COMPLIMENT)
                    1431            ;
036E AF             1432  SDADD:    XOR  A
036F 1A             1433  SDADD1:   LD   A,(DE)
0370 8E             1434            ADC  A,(HL)
0371 27             1435            DAA
0372 12             1436            LD   (DE),A
0373 13             1437            INC  DE
0374 23             1438            INC  HL
0375 10F8           1439            DJNZ SDADD1-$
0377 FE99           1440            CP   99H           ;
0379 17             1441            RLA                ;
037A 2F             1442            CPL                ;
037B FD7708         1443            LD   (IY+CBFLAG),A ;  SEND BACK STATUS FROM DADD
037E C9             1444            RET


                    1446  ; NAME:              RANGED RANDOM NUMBER
                    1447  ;  INPUT:            A = RANGE
                    1448  ;  OUTPUT:           A = RANDOM NUMBER (0 TO RANGE-1)
037F F5             1449  MRANGE:   PUSH AF
0380 2AEF4F         1450            LD   HL,(RANSHT)
0383 CDAC03         1451            CALL SHIFTR
0386 011700         1452            LD   BC,23
0389 09             1453            ADD  HL,BC
038A 8A             1454            ADC  A,D
038B 22EF4F         1455            LD   (RANSHT),HL
038E 2AF14F         1456            LD   HL,(RANSHT+2)
0391 5F             1457            LD   E,A
0392 CDAC03         1458            CALL SHIFTR
0395 19             1459            ADD  HL,DE
0396 22F14F         1460            LD   (RANSHT+2),HL
0399 5A             1461            LD   E,D
039A EB             1462            EX   DE,HL
039B F1             1463            POP  AF
039C A7             1464            AND  A
039D 4F             1465            LD   C,A
039E 7A             1466            LD   A,D
039F 2808           1467            JR   Z,R3-$
03A1 AF             1468            XOR  A
03A2 19             1469  R1:       ADD  HL,DE
03A3 3001           1470            JR   NC,R2-$
03A5 3C             1471            INC  A
03A6 0D             1472  R2:       DEC  C
03A7 20F9           1473            JR   NZ,R1-$
03A9 C3D10A         1474  R3:       JP   QFROG
03AC 44             1475  SHIFTR:   LD   B,H
03AD 4D             1476            LD   C,L
03AE AF             1477            XOR  A
03AF 1607           1478            LD   D,7
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

03B1 29        1479  SH1:     ADD  HL,HL
03B2 17        1480           RLA
03B3 15        1481           DEC  D
03B4 20FB      1482           JR   NZ,SH1-$
03B6 09        1483           ADD  HL,BC
03B7 8A        1484           ADC  A,D
03B8 C9        1485           RET


               1487  ; NAME:            SAVE AREA
               1488  ; INPUT:           HL = SCREEN ADDRESS
               1489  ;                  DE = SAVE AREA ADDRESS
               1490  ;                  BC = Y,X SIZE OF AREA TO SAVE
               1491  ; NOTES:           THE SIZES OF THE OBJECT ARE SAVED IN THE
               1492  ;                  FIRST TWO BYTES OF THE SAVE AREA.
03B9 EB        1493  MSAVE:   EX   DE,HL
03BA 71        1494           LD   (HL),C        ; SET X SIZE
03BB 23        1495           INC  HL
03BC 70        1496           LD   (HL),B        ; SET Y SIZE
03BD 23        1497           INC  HL
03BE AF        1498           XOR  A
03BF EB        1499           EX   DE,HL
03C0 CBF4      1500           SET  6,H           ; SET NONMAGIC ADDRESS
03C2 C5        1501  MSAVE1:  PUSH BC
03C3 E5        1502           PUSH HL
03C4 47        1503           LD   B,A
03C5 EDB0      1504           LDIR
03C7 E1        1505           POP  HL
03C8 0E28      1506           LD   C,BYTEPL
03CA 09        1507           ADD  HL,BC
03CB C1        1508           POP  BC
03CC 10F4      1509           DJNZ MSAVE1-$
03CE C9        1510           RET


               1512   ; NAME: PREGAME OUTPUT PORT SETUP
               1513   ; PURPOSE: TO SET CONCOM,VERBL ETC
               1514   ; INPUTS:  B=HORCB, D=VERBL, A=INMOD
03CF 0E09      1515  MSETUP:  LD   C,HORCB       ; GET BASE PORT NUMBER
03D1 ED41      1516           OUT  (C),B         ; HORBD
03D3 0C        1517           INC  C             ;
03D4 ED51      1518           OUT  (C),D         ; VERBL
03D6 D30E      1519           OUT  (INMOD),A
03D8 C9        1520           RET


               1522  ; NAME: TEST FOR TRANSITIONS
               1523  ; FUNCTION: TO LOOK FOR CHANGES IN THE PORTS &TC.
               1524  ; RETURNS : A= 0 NO CHANGE
               1525  ; 1-8 COUNTER TIMER#N HIT 0
               1526  ; 9-C = POTO-3 CHANGED
               1527  ; D = A SECONDS UP
               1528  ; E= KEYBOARD CHANGED (B=0-24)
               1529  ; F-16 : TRIGO!JOYO - T3!J3
               1530  ; RETURNS NEW VALUE IN B
03D9 5E        1531  CTLP    LD   E,(HL)
03DA 010108    1532          LD   BC,801H
03DD 79        1533  CCTLP   LD   A,C            ; GET MASK
03DE 0F        1534          RRCA
03DF 4F        1535          LD   C,A
03E0 A3        1536          AND  E              ; CHECK IF CT BIT =1
03E1 2003      1537          JR   NZ,CCT1-$
03E3 10F8      1538          DJNZ CCTLP-$
03E5 C9        1539          RET
03E6 AB        1540  CCT1:   XOR  E              ; MASK OUT BIT IN QUESTION
03E7 77        1541          LD   (HL),A         ; PUT BACK THE CTFLAGS OR SEMI4
03E8 78        1542          LD   A,B
03E9 82        1543          ADD  A,D
03EA E1        1544          POP  HL             ; OLD RET ADDR
03EB C9        1545          RET
03EC 2825      1546  TRCHK:  JR   Z,TSEX-$       ; SKIP COUNTER-TIMERS AND POTS?
03EE 21DD4F    1547          LD   HL,CNT         ; GET COUNTER   TIMERS STATUS
03F1 1600      1548          LD   D,0
03F3 CDD903    1549          CALL CTLP           ; COUNTER TIMERS
03F6 1608      1550          LD   D,8
03F8 23        1551          INC  HL
03F9 CDD903    1552          CALL CTLP           ; SEMI4S
03FC 011C04    1553          LD   BC,400H+POTO
03FF 23        1554  TPLOP   INC  HL             ; -> MPOTO
0400 ED78      1555          IN   A,(C)
0402 5E        1556          LD   E,(HL)         ; GET OPOT
0403 93        1557          SUB  E
0404 3805      1558          JR   C,PHOT-$       ; NEW ONE LESS THAN OLD
0406 D608      1559          SUB  PFUG           ; FUDGE. BOUNCE FACTOR
0408 3806      1560          JR   C,EPLOP-$      ; NEW MORE THAN OLD+4
040A 3C        1561          INC  A
040B 83        1562  PHOT:   ADD  A,E
040C 77        1563          LD   (HL),A
040D 47        1564          LD   B,A
040E 79        1565          LD   A,C
040F C9        1566          RET
0410 0C        1567  EPLOP   INC  C
0411 10EC      1568          DJNZ TPLOP-$
               1569  ; NOW TEST SECONDS
0413 21E34F    1570  TSEX:   LD   HL,KEYSEX      ; HL = KEYSEX
0416 7E        1571          LD   A,(HL)
0417 CB7F      1572          BIT  7,A
0419 2806      1573          JR   Z,TKEYS-$
041B CBBF      1574          RES  7,A
041D 77        1575          LD   (HL),A
041E 3E11      1576          LD   A,SSEC         ; SECS
0420 C9        1577          RET
               1578  ; NOW TEST KEYBOARD
0421 E5        1579  TKEYS:  PUSH HL
0422 CD7400    1580          CALL DELOAD
0425 EB        1581          EX   DE,HL
0426 011704    1582          LD   BC,400H+KEY3
0429 1100FF    1583          LD   DE,0FF00H      ; SET BIT COUNTER+COLUMNN
042C ED78      1584  MSK1:   IN   A,(C)
042E A6        1585          AND  (HL)           ; CHECK AGAINST MASK
042F 200A      1586          JR   NZ,MSENK2-$
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

0431 0D        1587           DEC  C              ; NEXT PORT
0432 1C        1588           INC  E              ; AND COLUMN
0433 23        1589           INC  HL             ; AND MASK
0434 10F6      1590           DJNZ MSK1-$
0436 78        1591           LD   A,B            ; NOTHING DOWN
0437 1E12      1592           LD   E,SKYU
0439 180B      1593           JR   MSENKE-$
043B 14        1594  MSENK2   INC  D              ; BIT COUNTER
043C 0F        1595           RRCA
043D 30FC      1596           JR   NC,MSENK2-$
043F 7A        1597           LD   A,D
0440 07        1598           RLCA                ; KEY=BIT*4
0441 07        1599           RLCA
0442 83        1600           ADD  A,E            ; + COLUMN
0443 3C        1601           INC  A              ; PLUS 1
0444 1E13      1602           LD   E,SKYD
0446 E1        1603  MSENKE   POP  HL
0447 AE        1604           XOR  (HL)           ; KEY=OKEY?
0448 E67F      1605           AND  7FH
044A 2807      1606           JR   Z,HANDLE-$
044C AE        1607           XOR  (HL)
044D 77        1608           LD   (HL),A
044E E67F      1609           AND  07FH
0450 47        1610           LD   B,A
0451 7B        1611           LD   A,E            ; KEYBOARD RETURN CODE
0452 C9        1612           RET
               1613  ; NOW TEST HANDLES
0453 011004    1614  HANDLE:  LD   BC,400H+SWO
0456 23        1615  SWLOP    INC  HL             ; -> OSWO
0457 ED78      1616           IN   A,(C)
0459 AE        1617           XOR  (HL)           ; COMPARE THE 2
045A 2005      1618           JR   NZ,SWHIT-$
045C 0C        1619           INC  C
045D 10F7      1620           DJNZ SWLOP-$        ; NO CHANGE
045F 78        1621           LD   A,B            ; RETURN 0
0460 C9        1622           RET
0461 CB67      1623  SWHIT:   BIT  4,A            ; TEST TRIGGER
0463 280C      1624           JR   Z,JOYS-$       ; NO TRIG MUST BE JOYSTICK
0465 E610      1625           AND  10H            ; FILTER OUT TRIGGER
0467 AE        1626           XOR  (HL)           ; UPDATE VALUE
0468 77        1627           LD   (HL),A
0469 E610      1628           AND  10H
046B 47        1629           LD   B,A
046C 79        1630           LD   A,C            ; GET PORT NUMBER
046D 07        1631           RLCA                ; *2
046E D60C      1632           SUB  0CH
0470 C9        1633           RET
0471 AE        1634  JOYS:    XOR  (HL)
0472 77        1635           LD   (HL),A         ; NO CHANGE IN TRIG SO STORE ST
0473 E60F      1636           AND  0FH            ; TAKE OFF TRIGGER
0475 47        1637           LD   B,A
0476 79        1638           LD   A,C
0477 07        1639           RLCA                ; *2
0478 D60B      1640           SUB  0BH
047A C9        1641           RET
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT


               1643  ; TIMEX
               1644  ; INPUTS HL-> TIME BASE IN RAM
               1645  ; B=TIME BASE MODULUS
               1646  ; C=MASK AS IN DECCTS
               1647  ; PURPOSE: TO DECR TIMEBASE AND IF 0 RESET IF AND DECR
               1648  ;  COUNTER       TIMERS
047B 35        1649  TIMEX:   DEC  (HL)           ; DEC TIMEBASE
047C C0        1650           RET  NZ
047D 70        1651           LD   (HL),B         ; RESET TIMEBASE


               1653  ; NAME: DECREMENT COUNTER        TIMERS
               1654  ; INPUTS: C=MASK
               1655  ; USED BY ACTINT AND DECCTS TO DECREMENTS CTS UNDER MASK
               1656  ; MASK= *76543210* , IF BIT=1 THEN DEC CORESPONDING
               1657  ; CT# , IF BIT=0 LEAVE CT# ALONE
               1658  ; NOTE: ALL COUNTERS ARE RUN IN BCD FOR EASY DISPLAY
047E 0608      1659  TIMEY:   LD   B,8            ; NO OF BITS
0480 21D54F    1660           LD   HL,CT0         ; -> TO COUNTER TIMERS
0483 1600      1661           LD   D,0            ; RESULTS
0485 CB39      1662  TIMLP:   SRL  C              ; CHANGE THIS TIMER?
0487 300A      1663           JR   NC,ETLP-$
0489 7E        1664           LD   A,(HL)         ; GET THE TIMER
048A B7        1665           OR   A              ; IS IT ZERO ALREADY?
048B 2806      1666           JR   Z,ETLP-$
048D 3D        1667           DEC  A
048E 27        1668           DAA
048F 2001      1669           JR   NZ,+3
0491 37        1670           SCF
0492 77        1671           LD   (HL),A         ; STORE NEW VALUE
0493 23        1672  ETLP:    INC  HL
0494 CB1A      1673           RR   D              ; ROTATES IN CARRY FLAG
0496 10ED      1674           DJNZ TIMLP-$
0498 3ADD4F    1675           LD   A,(CNT)        ; COUNTER UPDATE&NUMBER TRACKER
049B B2        1676           OR   D
049C 32DD4F    1677           LD   (CNT),A
049F C9        1678           RET


               1680   ; NAME: TIMER ROUTINE
               1681   ; PURPOSE: TO UPDATE GAME TIME,TIMOUT AND MUSIC
               1682  ; INPUTS         OUTPUTS: NONE
               1683  ; NOTE: PUSH YOUR REGISTERS (AF,BC,DE,HL)
               1684  TIMEZ:                       ; ASSUMES YOU PUSH DA REGS
04A0 21F94F    1685           LD   HL,PRIOR       ; PRIORITY=TICKS
04A3 CB4E      1686           BIT  1,(HL)         ;CHECK IF TICKS OVERRUN
04A5 C0        1687           RET  NZ             ; RETURN
04A6 CBCE      1688           SET  1,(HL)
04A8 EB        1689           EX   DE,HL
               1690    ;   *SIXTIETH OF A SECOND INTERUPT*
04A9 21EA4F    1691           LD   HL,DURAT       ; NOTE TIMER
04AC 7E        1692           LD   A,(HL)         ; =0 SKIP
04AD B7        1693           OR   A
```

```
ADDR OBJECT     STMT  LABEL    OPCD OPERAND        COMMENT

04AE 281C       1694           JR   Z,SIXY-$
04B0 35         1695           DEC  (HL)
04B1 200B       1696           JR   NZ,STAKO-$
04B3 E5         1697           PUSH HL
04B4 DDE5       1698           PUSH IX
04B6 CD1405     1699           CALL MUZCPU          ; =0 DO NEXT NOTE
04B9 DDE1       1700           POP  IX
04BB E1         1701           POP  HL
04BC 180E       1702           JR   SIXY-$
04BE EB         1703  STAKO:   EX   DE,HL
04BF CB7E       1704           BIT  7,(HL)
04C1 EB         1705           EX   DE,HL
04C2 2008       1706           JR   NZ,SIXY-$
04C4 3D         1707           DEC  A
04C5 3D         1708           DEC  A               ; =1 QUIET NOTE
04C6 2004       1709           JR   NZ,SIXY-$
                1710  ; A=0
04C8 D316       1711           OUT  (VOLAB),A
04CA D315       1712           OUT  (VOLC),A
04CC 23         1713  SIXY:    INC  HL
04CD 35         1714           DEC  (HL)            ;IF(--TMR60<0)
04CE F20205     1715           JP   P,GOUT          ; ELZ ONWARD
04D1 363B       1716           LD   (HL),59         ;THEN TMR60=59
04D3 23         1717           INC  HL              ; -> TIMOUT
04D4 EB         1718           EX   DE,HL
04D5 21E34F     1719           LD   HL,KEYSEX       ; SET SECONDS UP
04D8 CBFE       1720           SET  7,(HL)
04DA EB         1721           EX   DE,HL
04DB 7E         1722           LD   A,(HL)          ; CHECK IF ZERO
04DC B7         1723           OR   A
04DD 2801       1724           JR   Z,GTIMER-$
04DF 35         1725           DEC  (HL)            ; DEC TIMOUT
                1726    ; *GAME TIMER ONCE A SECOND ROUTINE*
                1727    ; IF (SEC != 0 & MIN !=0)
                1728    ;     IF (SEC == 0)
                1729    ;        SEC=59;--MIN
                1730    ;     ELSE --SEC
                1731    ; ELSE GAMETIMEUP=1
04E0 23         1732  GTIMER:  INC  HL              ;->GTSECS
04E1 7E         1733           LD   A,(HL)          ;IF (SEC!=0
04E2 23         1734           INC  HL              ;->GTMINS
04E3 B6         1735           OR   (HL)            ; & MIN!=0)
04E4 2813       1736           JR   Z,GT02-$
04E6 2B         1737           DEC  HL              ;->GTSECS AGAIN
04E7 7E         1738           LD   A,(HL)          ;IF (SEC ==0)
04E8 B7         1739           OR   A
04E9 2009       1740           JR   NZ,GT01-$
04EB 3659       1741           LD   (HL),59H        ; THEN SEC=59BCD
04ED 23         1742           INC  HL              ;->GTMINS AGAIN
04EE 7E         1743           LD   A,(HL)          ; --MIN
04EF 3D         1744           DEC  A
04F0 27         1745           DAA
04F1 77         1746           LD   (HL),A
04F2 180E       1747           JR   GOUT-$
04F4 3D         1748  GT01:    DEC  A               ; ELSE --SEC
04F5 27         1749           DAA
04F6 77         1750           LD   (HL),A
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

04F7 1809     1751          JR   GOUT-$
04F9 21F84F   1752  GT02:   LD   HL,GAMSTB    ; ELSE GAMETIMEUP=1
04FC CB46     1753          BIT  GSBTIM,(HL)
04FE 2802     1754          JR   Z,GOUT-$
0500 CBFE     1755          SET  GSBEND,(HL)
0502 21F94F   1756  GOUT    LD   HL,PRIOR
0505 CB8E     1757          RES  1,(HL)
0507 C9       1758          RET               ;RETURN TO BACKGND OR LO LEVEL
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

               1760   ; NAME: START MUZCPU
               1761   ; PURPOSE: TO START MUSIC PLAYING (ALSO NOISES)
               1762   ; INPUTS: HL -> SCORE
               1763   ; A=VOICES
               1764   ; NOTE: YOU SHOULD LOAD MUZSP IF YOU DO CALLS
0508 32D44F    1765  MUZSET  LD   (VOICES),A
050B DD22D04F  1766          LD   (MUZSP),IX
050F CDFC05    1767          CALL MUZSTP
0512 1803      1768          JR   MUZCP1-$
               1769   ; NAME: MUZCPU
               1770   ; PURPOSE: PLAYING MUSIC AND NOISES
               1771   ; NOTE: DURAT=0 WHEN CALLED
               1772   ; OUTPUT: NONE
               1773    ; *MUSIC PROCESSOR*
               1774    ;  FETCH OPCODE
               1775    ;  IF (OPCODE < 80H)
               1776    ;       SET NOTE DURATION ETC
               1777    ;  ELSE
               1778    ;  SWITCH (OPCODE & OFOH)
               1779    ;  CASE 80H:
               1780    ;       IF (MASK=8) STUFF SNDBX;PC=PC+9
               1781    ;       ELSE OUTPUT(MASK)=DATA
               1782    ;  CASE 90H:
               1783    ;       VOICES=DATA
               1784    ;  CASE AOH:
               1785    ;        (--SP)=DATA IN NIBBLE OF OP +1
               1786    ;  CASE BOH:
               1787    ;       SET VOLUMES = DATA,DATA
               1788    ;  CASE COH:
               1789    ;       SWITCH (MASK)
               1790    ;             CASE 9: MPCL=(MSP++); MPCH=(MSP++); BREAK
               1791    ;             CASE D: (--MSP)=MPCH; (--MSP)=MPCL
               1792    ;             CASE 0: IF --(SP)==0 THEN SP++
               1793    ;             CASE 3: MPC=DATA16
               1794    ;  CASE DOH: CALL RELATIVE
               1795   ;   CASE EO: DURAT=DATA
               1796    ;  CASE FO: VOICES=0,PORTS=0
0514 2ACE4F    1797  MUZCPU  LD   HL,(MUZPC)    ; LOOK LIKE NORMAL LOOP RETURN
0517 DD2AD04F  1798  MUZCP1  LD   IX,(MUZSP)    ; FETCH STACK POINTER
051B 7E        1799  OPLOOP  LD   A,(HL)        ; OPCODE FETCH
051C 23        1800          INC  HL            ;->OPERAND,DATA
051D B7        1801          OR   A             ; TEST FOR 80H OR MORE
051E FA5B05    1802          JP   M,MOO
               1803     ; NORMAL NOTE OPERATOR
0521 32EA4F    1804          LD   (DURAT),A
0524 3AD44F    1805          LD   A,(VOICES)
0527 011808    1806          LD   BC,800H+SNDBX
052A CB3F      1807          SRL  A             ; SET NOISE
052C 3002      1808          JR   NC,+4
052E EDA3      1809          OUTI
0530 0605      1810          LD   B,5           ; -> VIBRATO
0532 CB3F      1811          SRL  A
0534 3002      1812          JR   NC,+4
0536 EDA3      1813          OUTI               ; SET VIBRATO
0538 0604      1814          LD   B,4           ; -> NOTEC
053A CB3F      1815  M81:    SRL  A             ; CHECK C,B,A
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

053C 3009      1816           JR   NC,M82-$
053E EDA3      1817           OUTI
0540 CB3F      1818  M815     SRL  A               ; CHECK IF INC PC WAS ON
0542 3807      1819           JR   C,M83-$
0544 2B        1820           DEC  HL              ; RESTORE PC
0545 1804      1821           JR   M83-$
0547 05        1822  M82      DEC  B
0548 23        1823           INC  HL
0549 18F5      1824           JR   M815-$
054B B7        1825  M83      OR   A
054C 20EC      1826           JR   NZ,M81-$
               1827  ; PLAY NOTE
054E 3AD24F    1828           LD   A,(PVOLAB)
0551 D316      1829           OUT  (VOLAB),A
0553 3AD34F    1830           LD   A,(PVOLMC)
0556 D315      1831           OUT  (VOLC),A
0558 C3F405    1832           JP   MUZ999
055B FE90      1833  M00:     CP   90H
055D 3015      1834           JR   NC,M01-$
               1835     ; STUFF PORT OR SOUND BLOCK
055F CB5F      1836           BIT  3,A             ; IF (STUFF SNDBLK)
0561 2808      1837           JR   Z,M001-$
0563 78        1838           LD   A,B             ; SAVE B (VSN)
0564 011808    1839           LD   BC,8*256+SNDBX ; B=8,C=SNDBX
0567 EDB3      1840           OTIR                 ; HL->NEXT OPCODE WHEN DONE
0569 18B0      1841           JR   OPLOOP-$
056B E607      1842  M001:    AND  7               ; ISOLATE PORT NUMBER
056D F610      1843           OR   10H             ; PORTS 10H-17H
056F 4F        1844           LD   C,A             ; SET PORT REGISTER
0570 EDA3      1845           OUTI
0572 18A7      1846           JR   OPLOOP-$
0574 2007      1847  M01:     JR   NZ,M02-$
0576 7E        1848           LD   A,(HL)          ; GET NEW VOICES
0577 23        1849           INC  HL
0578 32D44F    1850           LD   (VOICES),A
057B 189E      1851           JR   OPLOOP-$
057D FEB0      1852  M02:     CP   0B0H
057F 3006      1853           JR   NC,M03-$
0581 E60F      1854           AND  0FH
0583 5F        1855           LD   E,A
0584 1C        1856           INC  E
0585 183E      1857           JR   M045-$
0587 FEC0      1858  M03:     CP   0C0H            ; SET VOL ETC
0589 3009      1859           JR   NC,M04-$
               1860  ; LOAD PVOLS
058B 11D24F    1861           LD   DE,PVOLAB
058E EDA0      1862           LDI                  ; DONT CARE ABOUT BC
0590 EDA0      1863           LDI
0592 1887      1864  OPLP2    JR   OPLOOP-$
0594 200B      1865  M04      JR   NZ,M040-$
0596 DD3500    1866           DEC  (IX+0)          ; DEC STACK TOP
0599 200A      1867           JR   NZ,M041-$
059B DD23      1868           INC  IX
059D 23        1869           INC  HL
059E 23        1870           INC  HL
059F 18F1      1871           JR   OPLP2-$
05A1 FED0      1872  M040     CP   0D0H            ; PC SP STUFF
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

05A3 3027      1873           JR   NC,M05-$
05A5 E60F      1874  M041     AND  0FH            ; ISOLATE MASK
05A7 FE09      1875           CP   9              ; RETURN
05A9 200C      1876           JR   NZ,M043-$
05AB DD6E00    1877           LD   L,(IX+0)
05AE DD23      1878           INC  IX
05B0 DD6600    1879           LD   H,(IX+0)
05B3 DD23      1880           INC  IX
05B5 18DB      1881           JR   OPLP2-$
05B7 5E        1882  M043:    LD   E,(HL)         ; PCL=
05B8 23        1883           INC  HL
05B9 56        1884           LD   D,(HL)         ; PCH=
05BA 23        1885           INC  HL
05BB EB        1886           EX   DE,HL          ; SET THE PC
05BC FE04      1887           CP   4              ; IS IT A JMP?
05BE 38D2      1888           JR   C,OPLP2-$      ; IT IS
05C0 DD2B      1889  M044     DEC  IX             ; ITS A CALL
05C2 DD7200    1890           LD   (IX+0),D       ; (--SP)=PCH
05C5 DD2B      1891  M045     DEC  IX
05C7 DD7300    1892           LD   (IX+0),E       ; (--SP)=PCL
05CA 18C6      1893           JR   OPLP2-$
05CC FEE0      1894  M05:     CP   0E0H
05CE 300A      1895           JR   NC,M06-$
05D0 E60F      1896           AND  0FH
05D2 0600      1897           LD   B,0
05D4 4F        1898           LD   C,A
05D5 54        1899           LD   D,H
05D6 5D        1900           LD   E,L
05D7 09        1901           ADD  HL,BC
05D8 18E6      1902           JR   M044-$         ; CALL
05DA 200A      1903  M06      JR   NZ,M061-$
05DC 3AF94F    1904           LD   A,(PRIOR)      ; LEGSTA
05DF EE80      1905           XOR  80H
05E1 32F94F    1906           LD   (PRIOR),A
05E4 18AC      1907           JR   OPLP2-$
05E6 FEF0      1908  M061     CP   0F0H           ; REST VOICE (OR SUSTAIN)
05E8 2812      1909           JR   Z,MUZSTP-$
05EA 7E        1910           LD   A,(HL)
05EB 32EA4F    1911           LD   (DURAT),A      ; SET DURATION OF QUIET
05EE 23        1912           INC  HL
05EF AF        1913           XOR  A
05F0 D316      1914           OUT  (VOLAB),A
05F2 D315      1915           OUT  (VOLC),A
               1916      ; END OF MUZIC PROCESSOR
05F4 22CE4F    1917  MUZ999:  LD   (MUZPC),HL     ; SAVE THE PC
05F7 DD22D04F  1918           LD   (MUZSP),IX     ; SAVE THE STACK POINTER
05FB C9        1919           RET
               1920   ; NAME MUZSTP
               1921   ; PURPOSE: STOP MUZCPU,SET PORTS TO 0
05FC AF        1922  MUZSTP:  XOR  A
05FD 32EA4F    1923           LD   (DURAT),A
0600 32F94F    1924           LD   (PRIOR),A
0603 011808    1925           LD   BC,800H+SNDBX
0606 ED79      1926           OUT  (C),A
0608 10FC      1927           DJNZ -2
060A C9        1928           RET
```

```
               1930  ; NAME:  DO IT
               1931  ; PURPOSE:       TRANSFER CONTROL TO USER STATE TRANSITION
               1932  ; INPUT:         A = RETURN CODE FROM SENTRY ROUTINE
               1933  ;          HL = DO IT TABLE ADDRESS
               1934  ; OUTPUT:
               1935  ; DESCRIPTION:   THIS ROUTINE IS USED WITH THE SENTRY ROUT
               1936  ;          IT IS USED FOR DISPATCHING TO A STATE TRANSITION
               1937  ;          ROUTINE.  THE RETURN CODE FROM SENTRY IS USED TO
               1938  ;          SEARCH THE DOIT TABLE.  IF A MATCH IS FOUND, CONT
               1939  ;          TRANSFERED.  IF NO MATCH IS FOUND, THE ROUTINE RE
               1940  ;          THE DOIT TABLE IS MADE UP OF THREE BYTE ENTRYS:
               1941  ;          BYTE 0 BIT 7:  IF SET - DO A MCALL TO THIS HANDLER
               1942  ;          BYTE 0 BIT 6:  IF SET - DO A RCALL TO THIS HANDLER
               1943  ;          BYTE 0 BITS 5-0:  RETURNCODE THIS ROUTINE IS TO PR
               1944  ;          BYTE 1 AND 2:  THE ADDRESS TO TRANSFER TO.
               1945  ;          THE LIST IS TERMINATED BY A BYTE WHICH IS .GE.  OC
060B 78        1946  MDOITB  LD   A,B
060C D5        1947  MDOIT:  PUSH DE
060D 57        1948          LD   D,A
060E 7E        1949  MDOIT0: LD   A,(HL)        ; GET RETURN CODE FOR THIS ENTR
060F 4F        1950          LD   C,A           ; C = CURRENT ENTRY
0610 FEC0      1951          CP   0C0H          ; LIST TERMINATOR?
0612 3802      1952          JR   C,MDOIT1-$    ; NO - JUMP
0614 D1        1953          POP  DE            ; YES - RETURN
0615 C9        1954          RET
0616 23        1955  MDOIT1: INC  HL
0617 E63F      1956          AND  3FH
0619 BA        1957          CP   D             ; NORMAL MATCH?
061A 2804      1958          JR   Z,MDOIT2-$    ; JUMP IF SO
061C 23        1959  MDO1A:  INC  HL            ; NO MATCH - SKIP OVER
061D 23        1960          INC  HL            ; GO TO ADDRESS
061E 18EE      1961          JR   MDOIT0-$
0620 D1        1962  MDOIT2: POP  DE
0621 5E        1963  MDOIT3: LD   E,(HL)        ; DE = GOTO ADDR
0622 23        1964          INC  HL
0623 56        1965          LD   D,(HL)
0624 EB        1966          EX   DE,HL
0625 CB79      1967          BIT  7,C           ; MCALL?
0627 C27D00    1968          JP   NZ,MMCALL     ; JUMP IF SO
062A CB71      1969          BIT  6,C           ; RCALL?
062C 2004      1970          JR   NZ,MRCALL-$
062E D1        1971          POP  DE            ; MUST BE JUMP
062F F1        1972          POP  AF
0630 E5        1973          PUSH HL
0631 EB        1974          EX   DE,HL
               1975  ; RCALL ROUTINE
0632 E9        1976  MRCALL: JP   (HL)
               1977   ; **********************
               1978   ; * VECTORING ROUTINES *
               1979   ; **********************
               1980   ; NAME:          VECTOR X AND Y COORDINATES
               1981   ; PURPOSE:       UPDATE X,Y COORDINATES AND LIMIT CHECK
               1982   ; INPUT:         IX = VECTOR PACKET
               1983   ;                HL = LIMITS TABLE
               1984   ; OUTPUT:        C = TIME BASE USED
               1985   ;                NONZERO STATUS SET IF OBJECT MOVED
```

```
ADDR OBJECT     STMT  LABEL    OPCD OPERAND        COMMENT

                1986   ; NOTES:
                1987   ;   THIS ROUTINE WORKS WITH A 'VECTOR PACKET', WHICH LOO
                1988   ; *********************************
                1989   ; *BYTE*    CONTENTS      * NAME    *
                1990   ; *********************************
                1991   ; * 00 * MAGIC REGISTER * VBMR    *
                1992   ; *********************************
                1993   ; * 01 * VECTOR STATUS  * VBSTAT *
                1994   ; *********************************
                1995   ; * 02 * TIME BASE      * VBTIMB *
                1996   ; *********************************
                1997   ; * 03 * DELTA X        * VBDXL   *
                1998   ; * 04 *                * VBDXH   *
                1999   ; *********************************
                2000   ; * 05 * X COORDINATE   * VBXL    *
                2001   ; * 06 *                * VBXH    *
                2002   ; *********************************
                2003   ; * 07 * X CHECKS MASK  * VBXCHK *
                2004   ; *********************************
                2005   ; * 08 * DELTA Y        * VBDYL   *
                2006   ; * 09 *                * VBDYH   *
                2007   ; *********************************
                2008   ; * 0A * Y COORDINATE   * VBYL    *
                2009   ; * 0B *                * VBYH    *
                2010   ; *********************************
                2011   ; * 0C * Y CHECKS MASK  * VBYCHK *
                2012   ; *********************************
                2013   ;
                2014   ; OPTIONS BYTE:
                2015   ; BIT  MEANING
                2016   ; ---  -------
                2017   ;  7   VECTOR IS ACTIVE
                2018   ;
                2019   ; CHECKS BYTE:
                2020   ; BIT  MEANING
                2021   ; ---  -------
                2022   ;  0   DO LIMIT CHECKS
                2023   ;  1   REVERSE COORDINATES ON LIMIT ATTAINMENT
                2024   ;  3   TARGET ATTAINED (OUTPUT)
                2025   ;   IF THE VECTOR IS ACTIVE, AND THE TIME BASE IS NONZER
                2026   ; THEN THE UPDATE COORDINATE ROUTINE IS CALLED FOR THE X
                2027   ; AND Y PORTIONS OF THE PACKET.
0633 FDCB08F6   2028  MVECT:   SET  PSWZRO,(IY+CBFLAG) ; SET ZERO FLAG
0637 DDCB017E   2029           BIT  VBSACT,(IX+VBSTAT) ; IS VECTOR ACTIVE?
063B DD4E02     2030           LD   C,(IX+VBTIMB) ; TIME BASE TO C
063E DD360200   2031           LD   (IX+VBTIMB),0 ; ZERO TIME BASE
0642 FD7106     2032           LD   (IY+CBC),C    ; PASS BACK TIME BASE
0645 C8         2033           RET  Z
0646 79         2034           LD   A,C
0647 A7         2035           AND  A              ; IS TIME BASE ZERO?
0648 C8         2036           RET  Z              ; QUIT IF SO
0649 110300     2037           LD   DE,VBDXL       ; ADVANCE TO FIRST
064C DD19       2038           ADD  IX,DE
064E CD5606     2039           CALL MVECTC         ; UPDATE FIRST COORDINATE
0651 110500     2040           LD   DE,VBDYL-VBDXL ; TO Y
0654 DD19       2041           ADD  IX,DE
                2042  ; AND FALL INTO ...
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

               2043   ; NAME:          VECTOR COORDINATE
               2044   ; PURPOSE:       UPDATE OF SINGLE COORDINATE
               2045   ; INPUT:         IX = POINTER TO L.O. DELTA BYTE OF VECTOR
               2046   ;                C = TIME BASE
               2047   ;                HL = LIMITS PACKET (IF USED)
               2048   ; OUTPUT:        NONZERO STATUS SET IF MOTION OCCURED
               2049   ;                (SHOULD BE SET ON CALL, SINCE IT IS NOT S
               2050   ; NOTES:
               2051   ;   THIS ROUTINE OPERATES ON A SUBSET OF THE VECTOR PACK
               2052   ; (BETWEEN L.O. DELTA BYTE AND CHECKS BYTE).
               2053   ;   THE DELTA IS ADDED TO THE COORDINATE TIME-BASE TIMES
               2054   ; IF OPTIONED, LIMIT CHECKING IS DONE.  IF THE CHECK FAI
               2055   ; THE COORDINATE IS SET TO THE LIMIT.
               2056   ; WHEN THIS HAPPENS, THE LIMIT ATTAINED BIT IS SET
0656 E5        2057  MVECTC: PUSH HL
0657 DD5601    2058          LD   D,(IX+VBDCH) ; LOAD DELTA
065A DD5E00    2059          LD   E,(IX+VBDCL)
065D DD6603    2060          LD   H,(IX+VBCH)  ; LOAD COORDINATE
0660 DD6E02    2061          LD   L,(IX+VBCL)
0663 7C        2062          LD   A,H          ; SAVE OLD COORDINATE FOR MOTIO
0664 41        2063          LD   B,C
0665 19        2064  MVECT1: ADD  HL,DE        ; ADD DELTA TO COORD
0666 10FD      2065          DJNZ MVECT1-$     ; TIME-BASE TIMES
               2066    ; HAS MOTION OCCURED?
0668 BC        2067          CP   H
0669 2804      2068          JR   Z,MVCT1A-$   ; JUMP TO SKIP TESTS IF SO
066B FDCB08B6  2069          RES  PSWZRO,(IY+CBFLAG) ; SET MOVED STATUS
               2070    ; IS LIMIT CHECK WANTED?
066F DDCB0446  2071  MVCT1A: BIT  VBCLMT,(IX+VBCCHK)
0673 2831      2072          JR   Z,MVECT6-$   ; MVECT6 IF NOT
               2073    ; PERFORM LIMIT CHECK
0675 7C        2074          LD   A,H
0676 E3        2075          EX   (SP),HL
0677 46        2076          LD   B,(HL)       ; LIMIT TO B
0678 23        2077          INC  HL
               2078  ; HANDLE SLIGHTLY LESS THAN ZERO CASE
0679 FECF      2079          CP   207          ; MIDPOINT BETWEEN 160 AND 0
067B 3007      2080          JR   NC,MVECT2-$  ; JUMP TO FAIL IF >207
067D B8        2081          CP   B            ; DO COMPARE
067E 3804      2082          JR   C,MVECT2-$   ; JUMP ON FAIL
0680 46        2083          LD   B,(HL)       ; UPPER LIMIT CHECK
0681 B8        2084          CP   B
0682 3820      2085          JR   C,MVECT3-$   ; JUMP ON PASS
0684 23        2086  MVECT2: INC  HL
               2087  ; A LIMIT WAS EXCEEDED - SET COORDINATE AT LIMIT
0685 DD7003    2088          LD   (IX+VBCH),B
0688 DD360200  2089          LD   (IX+VBCL),0
068C DDCB04DE  2090          SET  VBCLAT,(IX+VBCCHK) ; SET LIMIT ATTAINED
               2091  ; IS REVERSE DELTA OPTION SET?
0690 F1        2092          POP  AF           ; CLEAN UP STACK
0691 DDCB044E  2093          BIT  VBCREV,(IX+VBCCHK)
0695 C8        2094          RET  Z            ; QUIT IF NOT
               2095  ; REVERSE THE BIMBO
0696 7A        2096          LD   A,D
0697 2F        2097          CPL
0698 57        2098          LD   D,A
0699 7B        2099          LD   A,E
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

069A 2F        2100           CPL
069B 5F        2101           LD   E,A
069C 13        2102           INC  DE
069D DD7300    2103           LD   (IX+VBDCL),E ;  STORE BACK
06A0 DD7201    2104           LD   (IX+VBDCH),D
06A3 C9        2105           RET
06A4 23        2106  MVECT3:  INC  HL              ;  STEP PAST LIMIT
06A5 E3        2107           EX   (SP),HL         ;  HL = COORDINATE AGAIN
06A6 DD7502    2108  MVECT6:  LD   (IX+VBCL),L    ;  STORE BACK COORDINATES
06A9 DD7403    2109           LD   (IX+VBCH),H
06AC E1        2110           POP  HL              ;  RESTORE LIMITS POINTER
06AD DDCB049E  2111           RES  VBCLAT,(IX+VBCCHK) ;  CLEAR ATTAINED BIT
06B1 C9        2112           RET
```

```
               2114  ; **************************
               2115  ; * PAINT RECTANGLE ROUTINE *
               2116  ; **************************
               2117  ; NAME:               PAINT RECTANGLE
               2118  ; INPUT:              A = COLOR MASK TO WRITE
               2119  ;                     B = Y SIZE
               2120  ;                     C = X SIZE
               2121  ;                     D = Y COORDINATE
               2122  ;                     E = X COORDINATE
06B2 AF        2123  MPAINT:  XOR  A
06B3 CD4E0B    2124           CALL RELTA1
06B6 EB        2125           EX   DE,HL
06B7 CBF4      2126           SET  6,H             ; UNMAGIC THE ADDRESS
06B9 D30C      2127           OUT  (MAGIC),A
               2128  ;        XOR     A
06BB FD5E09    2129           LD   E,(IY+CBA)
06BE 79        2130           LD   A,C
06BF 0F        2131           RRCA
06C0 0F        2132           RRCA
06C1 E63F      2133           AND  3FH
06C3 3C        2134           INC  A
06C4 57        2135           LD   D,A
06C5 15        2136  MPT1:    DEC  D
06C6 2807      2137           JR   Z,MPT2-$
06C8 3EFF      2138           LD   A,0FFH
06CA CDE206    2139           CALL STRIPE
06CD 18F6      2140           JR   MPT1-$
06CF 79        2141  MPT2:    LD   A,C
06D0 E603      2142           AND  03H
06D2 3C        2143           INC  A
06D3 4F        2144           LD   C,A
06D4 AF        2145           XOR  A
06D5 0D        2146  MPT3:    DEC  C
06D6 2806      2147           JR   Z,MPT4-$
06D8 0F        2148           RRCA
06D9 0F        2149           RRCA
06DA C6C0      2150           ADD  A,11000000B
06DC 18F7      2151           JR   MPT3-$
06DE CDE206    2152  MPT4:    CALL STRIPE
06E1 AF        2153           XOR  A
               2154  ; AND FALL INTO . . .
               2155  ; STRIPE PAINTER
               2156  ; HL = ADDRESS OF STRIPE A = DATA E =MASK B = ITERATIONS
               2157  ; OUT HL=HL+1 A = CLOBBERED
06E2 E5        2158  STRIPE:  PUSH HL
06E3 C5        2159           PUSH BC
06E4 32FF0F    2160           LD   (WASTE),A
06E7 3AFF4F    2161           LD   A,(WASTE+4000H)
06EA 4F        2162           LD   C,A
06EB 7B        2163  STRP1:   LD   A,E
06EC AE        2164           XOR  (HL)
06ED A1        2165           AND  C
06EE AE        2166           XOR  (HL)
06EF 77        2167           LD   (HL),A
06F0 7D        2168           LD   A,L
06F1 C628      2169           ADD  A,BYTEPL
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

06F3 6F       2170          LD   L,A
06F4 7C       2171          LD   A,H
06F5 CE00     2172          ADC  A,0
06F7 67       2173          LD   H,A
06F8 10F1     2174          DJNZ STRP1-$
06FA C1       2175          POP  BC
06FB E1       2176          POP  HL
06FC 23       2177          INC  HL
06FD C9       2178          RET
```

```
              2180   ; ******************
              2181   ; * WRITE ROUTINES *
              2182   ; ******************
              2183   ; NOTES:          THE GENERAL CALLING SEQUENCE FOR THE WRI
              2184   ; INPUT:          HL = PATTERN ADDRESS
              2185   ;                 D = Y COORDINATE
              2186   ;                 E = X COORDINATE
              2187   ;                 B = Y SIZE
              2188   ;                 C = X SIZE
              2189   ;                 A = MAGIC REGISTER
              2190   ; OUTPUT:         DE = SCREEN ADDRESS USED
              2191   ;                   THESE ROUTINES ARE NESTED, FOR EXAMPLE
              2192   ;                 WRITP, WHICH FALLS INTO WRIT, WHICH FALL
              2193   ; ENTRY:          WRITE FROM VECTOR
              2194   ; INPUT:          HL = PATTERN ADDRESS
              2195   ;                 IX = VECTOR ADDRESS
              2196   ; OUTPUT:         DE,A
              2197   ; SIDE EFFECTS:  BLANK BIT SET IN VECTOR STATUS BYTE
06FE DD7E00   2198  MVWRIT: LD   A,(IX+VBMR)   ; LOAD MR
0701 DD560B   2199          LD   D,(IX+VBYH)   ; LOAD Y
0704 DD5E06   2200          LD   E,(IX+VBXH)   ; LOAD X
0707 DDCB01F6 2201          SET  VBBLNK,(IX+VBSTAT) ; SET BLANK BIT
              2202   ; ENTRY:          WRITE RELATIVE
              2203   ; PURPOSE:        WRITING RELATIVE PATTERNS
              2204   ; INPUT:          HL,DE,A
              2205   ; OUTPUT:         DE
              2206   ; NOTES:          PATTERN IS PRECEEDED BY RELATIVE DISPLAC
              2207   ;                 (X FIRST, THEN Y) AND PATTERN SIZE
070B F5       2208  MWRITR: PUSH AF            ; SAVE MR
070C 7E       2209          LD   A,(HL)        ; GET REL X
070D 23       2210          INC  HL
070E 83       2211          ADD  A,E           ; ADD TO SUPERIOR X
070F 5F       2212          LD   E,A
0710 7E       2213          LD   A,(HL)        ; SAME STORY FOR Y
0711 23       2214          INC  HL
0712 82       2215          ADD  A,D
0713 57       2216          LD   D,A
0714 F1       2217          POP  AF
              2218   ; ENTRY:          WRITE WITH PATTERN SIZE SCARE-UP
              2219   ; PURPOSE:        WRITING VARIABLE SIZED PATTERNS
              2220   ; INPUT:          HL,DE,A
              2221   ; OUTPUT:         DE
              2222   ; NOTES:          FIRST TWO BYTES POINTED AT BY HL ARE TAK
              2223   ;                 TO BE PATTERN SIZES (X SIZE FIRST)
0715 4E       2224  MWRITP: LD   C,(HL)        ; GET X SIZE
0716 23       2225          INC  HL
0717 46       2226          LD   B,(HL)        ; AND Y
0718 23       2227          INC  HL
              2228   ; ENTRY:          WRITE WITH COORDINATE CONVERSION
              2229   ; INPUT:          HL,DE,BC,A
              2230   ; OUTPUT:         DE
0719 CDF60A   2231  MWRIT:  CALL MRELAB        ; DO CONVERSION
              2232   ; ENTRY:          WRITE ABSOLUTE
              2233   ; INPUT:          HL,BC,A AS ABOVE
              2234   ;                 DE = ABSOLUTE SCREEN ADDRESS
071C CB77     2235  MWRITA: BIT  MRFLOP,A      ; FLOP WRITE WANTED?
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

071E 202C     2236          JR   NZ,MWRTFL-$   ; MWRTFL IF SO
0720 CB5F     2237          BIT  MRXPND,A      ; EXPAND WANTED?
0722 2011     2238          JR   NZ,MWX-$      ; JUMP IF SO
              2239    ; DO NORMAL? WRITE
0724 AF       2240          XOR  A
0725 C5       2241  MWRT:   PUSH BC
0726 D5       2242          PUSH DE
0727 47       2243          LD   B,A           ; ZERO REGISTER B
0728 EDB0     2244          LDIR               ; WRITE A LINE
072A 12       2245          LD   (DE),A        ; CLEAR THE SHIFTER
072B D1       2246          POP  DE
072C EB       2247          EX   DE,HL         ; ADVANCE TO NEXT LINE
072D 0E28     2248          LD   C,BYTEPL
072F 09       2249          ADD  HL,BC
0730 EB       2250          EX   DE,HL
0731 C1       2251          POP  BC
0732 10F1     2252          DJNZ MWRT-$        ; LOOP IF MORE GOODIES
0734 C9       2253          RET
              2254  ; WRITE EXPANDED
0735 EB       2255  MWX:    EX   DE,HL
0736 C5       2256  MWX1:   PUSH BC
0737 E5       2257          PUSH HL
0738 41       2258          LD   B,C
0739 1A       2259  MWX2:   LD   A,(DE)
073A 13       2260          INC  DE
073B 77       2261          LD   (HL),A
073C 23       2262          INC  HL
073D 77       2263          LD   (HL),A
073E 23       2264          INC  HL
073F 10F8     2265          DJNZ MWX2-$
0741 70       2266          LD   (HL),B
0742 23       2267          INC  HL
0743 70       2268          LD   (HL),B
0744 E1       2269          POP  HL
0745 0E28     2270          LD   C,BYTEPL
0747 09       2271          ADD  HL,BC
0748 C1       2272          POP  BC
0749 10EB     2273          DJNZ MWX1-$
074B C9       2274          RET
              2275    ; ROUTINE TO HANDLE FLOPPED CASE
074C CB5F     2276  MWRTFL: BIT  MRXPND,A      ; EXPANDED FLOPPED WRITE WANTED
074E 2016     2277          JR   NZ,MWXF-$     ; JUMP IF YEP
0750 AF       2278          XOR  A
0751 C5       2279  WRFL1:  PUSH BC
0752 D5       2280          PUSH DE
0753 47       2281          LD   B,A
0754 EDA0     2282  WRFL2:  LDI
0756 1B       2283          DEC  DE
0757 1B       2284          DEC  DE
0758 EA5407   2285          JP   PE,WRFL2
075B 12       2286          LD   (DE),A        ;
075C D1       2287          POP  DE
075D EB       2288          EX   DE,HL         ; SAME AS NORMAL NOW ON
075E 0E28     2289          LD   C,BYTEPL
0760 09       2290          ADD  HL,BC
0761 EB       2291          EX   DE,HL
0762 C1       2292          POP  BC
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

0763 10EC      2293           DJNZ WRFL1-$
0765 C9        2294           RET
               2295  ; WRITE EXPANDED FLOPPED ROUTINE
0766 EB        2296  MWXF:    EX   DE,HL
0767 C5        2297  MWXF1:   PUSH BC
0768 E5        2298           PUSH HL
0769 41        2299           LD   B,C
076A 1A        2300  MWXF2:   LD   A,(DE)
076B 13        2301           INC  DE
076C 77        2302           LD   (HL),A
076D 2B        2303           DEC  HL
076E 77        2304           LD   (HL),A
076F 2B        2305           DEC  HL
0770 10F8      2306           DJNZ MWXF2-$
0772 70        2307           LD   (HL),B
0773 2B        2308           DEC  HL
0774 70        2309           LD   (HL),B
0775 E1        2310           POP  HL
0776 0E28      2311           LD   C,BYTEPL
0778 09        2312           ADD  HL,BC
0779 C1        2313           POP  BC
077A 10EB      2314           DJNZ MWXF1-$
077C C9        2315           RET
               2316   ; NAME:          BLANK FROM VECTOR
               2317   ; PURPOSE:       BLANK WITH INFO LOAD FROM VECTOR
               2318   ; INPUT:         IX = VECTOR
               2319   ;              E = X SIZE
               2320   ;              D = Y SIZE
               2321   ; NOTES:         THIS ROUTINE BLANKS TO 00
               2322   ;              THIS ROUTINE INTERROGATES THE BLANK BIT
               2323   ;              AND REFRAINS FROM BLANKING IF NOT SET
               2324   ;              IF IT WAS SET, IT IS THEN RESET
077D DDCB0176  2325  MVBLAN:  BIT  VBBLNK,(IX+VBSTAT) ; IS BLANK BIT SET?
0781 C8        2326           RET  Z                ; QUIT IF NOT
0782 DDCB01B6  2327           RES  VBBLNK,(IX+VBSTAT) ; KILL BLANK BIT
0786 DD660E    2328           LD   H,(IX+VBOAH) ; LOAD BLANK ADDRESS
0789 DD6E0D    2329           LD   L,(IX+VBOAL)
078C DDCB0076  2330           BIT  MRFLOP,(IX+VBMR) ; IS FLOP SET?
0790 2808      2331           JR   Z,MVBLA1-$    ; JUMP IF NOT
0792 7B        2332           LD   A,E             ; X SIZE TO A
0793 ED44      2333           NEG                  ; TWOS COMPLEMENT AND ADD 1
0795 3C        2334           INC  A
0796 4F        2335           LD   C,A
0797 06FF      2336           LD   B,0FFH
0799 09        2337           ADD  HL,BC           ; USE TO BACK UP SCREEN ADDRESS
               2338  ; UNMAGIC THE BLANK ADDRESS
079A           2339  MVBLA1:
079A CBF4      2340           SET  6,H
079C 0600      2341           LD   B,0             ; ASSUME BLANK TO ZERO
               2342   ; NAME:          BLANK AREA
               2343   ; PURPOSE:       SETTING N X M REGION TO CONSTANT
               2344   ; INPUT:         HL = BLANK ADDRESS
               2345   ;              E = X SIZE
               2346   ;              D = Y SIZE
               2347   ;              B = DATA TO FILL WITH
079E 3E28      2348  MBLANK:  LD   A,BYTEPL      ; COMPUTE LINE INCREMENT
07A0 93        2349           SUB  E
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

07A1 4F       2350          LD   C,A
07A2 78       2351          LD   A,B          ; A = DATA TO FILL WITH
07A3 43       2352  MBLAN1: LD   B,E
07A4 77       2353  MBLAN2: LD   (HL),A
07A5 23       2354          INC  HL
07A6 10FC     2355          DJNZ MBLAN2-$
07A8 09       2356          ADD  HL,BC
07A9 15       2357          DEC  D
07AA 20F7     2358          JR   NZ,MBLAN1-$
07AC C9       2359          RET
              2360  ; NAME:            RESTORE AREA
              2361  ; INPUT:           HL = SCREEN ADDRESS TO RESTORE TO
              2362  ;                  DE = SAVE AREA ADDRESS
              2363  ; NOTE:            SIZES ARE LOADED FROM THE SAVE AREA
07AD EB       2364  MREST:  EX   DE,HL
07AE 4E       2365          LD   C,(HL)
07AF 23       2366          INC  HL
07B0 46       2367          LD   B,(HL)
07B1 23       2368          INC  HL
07B2 CBF2     2369          SET  6,D          ; MAKE SURE WE ARE NONMAGIC
07B4 AF       2370          XOR  A
07B5 C5       2371  MREST1: PUSH BC
07B6 D5       2372          PUSH DE
07B7 47       2373          LD   B,A
07B8 EDB0     2374          LDIR
07BA EB       2375          EX   DE,HL
07BB E1       2376          POP  HL
07BC 0E28     2377          LD   C,BYTEPL
07BE 09       2378          ADD  HL,BC
07BF EB       2379          EX   DE,HL
07C0 C1       2380          POP  BC
07C1 10F2     2381          DJNZ MREST1-$
07C3 C9       2382          RET
```

```
                2384  ; ******************************
                2385  ; * CHARACTER DISPLAY ROUTINES *
                2386  ; ******************************
                2387  ; NAME:          DISPLAY STRING
                2388  ; PURPOSE:       MESSAGE DISPLAY
                2389  ; INPUT:         E,D = X, Y COORDINATES
                2390  ;                HL = STRING ADDRESS
                2391  ;                IX = FONT DESCRIPTOR
                2392  ; OUTPUT:        D,E ALTERED AS IN DISPLAY CHARACTER
                2393  ; STACK USE:     4 BYTES (EXCLUDING USE BY SYSPCH)
                2394  ; EXPLANATION: AS EACH CHARACTER IS BROUGHT IN, IT
                2395  ; IS TESTED FOR BEING A LIST TERMINATOR ( CHAR = 0)
                2396  ; IF IT ISN'T, DISPLAY CHARACTER IS CALLED AND THE
                2397  ; TEST IS REPEATED FOR THE NEXT CHARACTER.  THUS
                2398  ; A NULL STRING IS HANDLED PROPERLY.
07C4 7E         2399 STRNEW: LD   A,(HL)         ; GET CHARACTER
07C5 A7         2400         AND  A              ; BE IT A TERMINATOR?
07C6 C8         2401         RET  Z              ; QUIT IF SO
07C7 FACE07     2402         JP   M,STRD1        ; DISPLAY IF ALT FONT
07CA FE64       2403         CP   64H            ; SUCK IN STRING?
07CC 3006       2404         JR   NC,STRD2-$     ; JUMP IF YES
07CE CDE107     2405 STRD1:  CALL DISPCH         ; SHOW CHAR
07D1 23         2406         INC  HL             ; ADVANCE TO NEXT CHAR
07D2 18F0       2407         JR   STRNEW-$       ; AND LOOP
07D4 E617       2408 STRD2:  AND  10111B         ; MAKE SUCK MASK
07D6 47         2409         LD   B,A
07D7 23         2410         INC  HL
07D8 EB         2411         EX   DE,HL
07D9 CDA800     2412         CALL MSUCK1
07DC CD6800     2413         CALL RELD
07DF 18E3       2414         JR   STRNEW-$       ; GO AFTER NEXT CHARACTER
                2415 ; *****************************
                2416 ; * CHARACTER DISPLAY ROUTINE *
                2417 ; *****************************
                2418 ; INPUT:          A = CHARACTER
                2419 ;                 C = OPTIONS
                2420 ;                 D = Y COORDINATE
                2421 ;                 E = X COORDINATE
                2422 ;                 IX = FONT DESCRIPTOR
                2423 ;                 (ONLY IF ALTERNATE FONT USED)
                2424 ; OUTPUT:         DE UPDATED TO POINT AT NEXT CHARACTER FRA
                2425 ; NOTES:          THE OPTION BYTE IS FORMATTED AS FOLLOWS:
                2426 ;          BITS   CONTENTS
                2427 ;          ----   --------
                2428 ;          0-1    OFF COLOR FOR EXPANSION
                2429 ;          2-3    ON COLOR FOR EXPANSION
                2430 ;          4      OR OPTION
                2431 ;          5      XOR OPTION
                2432 ;          6-7    ENLARGEMENT FACTOR (N+1)X
                2433 ;
                2434 ; CHARACTERS BETWEEN 1 AND 1FH, AND BETWEEN 81H AND 9FH
                2435 ; ARE INTERPRETED AS TAB CHARACTERS.  THEY CAUSE THE
                2436 ; CURSOR REPRESENTED BY D AND E TO BE SPACED OVER N
                2437 ; CHARACTER POSITIONS, WHERE N = CHAR. AND. 7FH
                2438 ; CHARACTERS BETWEEN 20H AND 7FH ARE TAKEN AS REFERENCES
                2439 ; THE SYSTEM STANDARD 5 X 7 CHARACTER FONT.  CHARACTERS
```

```
              2440  ; BETWEEN 0A0H AND 0FFH REFER TO THE USER SUPPLIED ALTERN
              2441  ; CHARACTER FONT.  THIS FONT IS DESCRIBED BY A FONT
              2442  ; DESCRIPTOR TABLE OF THE FOLLOWING FORMAT:
              2443  ; ******************************
              2444  ; * 0 * BASE CHARACTER VALUE   *
              2445  ; ******************************
              2446  ; * 1 * X FRAME SIZE           *
              2447  ; ******************************
              2448  ; * 2 * Y FRAME SIZE           *
              2449  ; ******************************
              2450  ; * 3 * X PATTERN SIZE (BYTES) *
              2451  ; ******************************
              2452  ; * 4 * Y PATTERN SIZE         *
              2453  ; ******************************
              2454  ; * 5 *   PATTERN TABLE        *
              2455  ; * 6 *      ADDRESS           *
              2456  ; ******************************
07E1 C5       2457  DISPCH: PUSH BC
07E2 E5       2458          PUSH HL
07E3 DDE5     2459          PUSH IX
07E5 A7       2460          AND  A
07E6 FAED07   2461          JP   M,DISCH1       ; JUMP IF YES
07E9 DD210602 2462          LD   IX,SYSFNT
07ED FE20     2463  DISCH1: CP   20H            ; IS CHAR < 20H?
07EF 300D     2464          JR   NC,DISC1B-$    ; JUMP IF NOT
07F1 F5       2465  DISC1A: PUSH AF             ; LOOP TO SPACE OVER
07F2 CD4E08   2466          CALL NXTFRM
07F5 CDF40C   2467          CALL FINDL3         ; STORE IT BACK
07F8 F1       2468          POP  AF
07F9 3D       2469          DEC  A
07FA 20F5     2470          JR   NZ,DISC1A-$
07FC 183B     2471          JR   DISCH5-$       ; JUMP TO EXIT
07FE DD9600   2472  DISC1B: SUB  (IX+FTBASE)    ; SUBTRACT BASE CHAR
0801 5F       2473          LD   E,A
0802 1600     2474          LD   D,0
0804 210000   2475          LD   HL,0
0807 DD4E03   2476          LD   C,(IX+FTBYTE) ; MULTIPLY CHARACTER
080A DD4604   2477  DISCH2: LD   B,(IX+FTYSIZ) ; BY PATTERN SIZE
080D 19       2478  DISCH3: ADD  HL,DE
080E 10FD     2479          DJNZ DISCH3-$
0810 0D       2480          DEC  C
0811 20F7     2481          JR   NZ,DISCH2-$
0813 DD5606   2482          LD   D,(IX+FTPTH) ; ADD TO TABLE START
0816 DD5E05   2483          LD   E,(IX+FTPTL)
0819 19       2484          ADD  HL,DE
              2485  ; COMPUTE POSITION WHERE NEXT CHARACTER WOULD GO
              2486  ; AND SAVE
081A CD4E08   2487          CALL NXTFRM         ; STEP COORDINATES TO NEXT FRAM
081D D5       2488          PUSH DE             ; SAVE
081E DD4604   2489          LD   B,(IX+FTYSIZ)
0821 C5       2490  DISCH4: PUSH BC
0822 E5       2491          PUSH HL
0823 CD6C08   2492          CALL WRTLIN
0826 E1       2493          POP  HL
0827 DD4E03   2494          LD   C,(IX+FTBYTE) ; STEP TO NEXT LINE OF PATTERN
082A 09       2495          ADD  HL,BC
082B C1       2496          POP  BC
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

082C FD7E05   2497          LD   A,(IY+CBD)    ; ADVANCE Y COORDINATE
082F 81       2498          ADD  A,C
0830 FD7705   2499          LD   (IY+CBD),A
0833 10EC     2500          DJNZ DISCH4-$
0835 D1       2501          POP  DE            ; RESTORE NEW POSITION
0836 CDF40C   2502          CALL FINDL3        ; STUFF DE BACK INTO CONTEXT
0839 DDE1     2503  DISCH5: POP  IX
083B E1       2504          POP  HL
083C C1       2505          POP  BC
083D C9       2506          RET
              2507  ; SUBROUTINE TO CONVERT ENLARGEMENT FACTOR TO ITERATION C
              2508  ; INPUT:        MODE BYTE FROM CONTEXT SAVE AREA
              2509  ; OUTPUT:       B,A = ITERATION COUNT
083E FD7E06   2510  DCLCTB: LD   A,(IY+CBC)    ; GET MODE BYTE
0841 07       2511          RLCA
0842 07       2512          RLCA
0843 E603     2513          AND  03            ; ISOLATE ENLARGEMENT FACTOR
0845 3C       2514          INC  A
0846 47       2515          LD   B,A
0847 AF       2516          XOR  A
0848 37       2517          SCF
0849 8F       2518  DCLCT1: ADC  A,A
084A 10FD     2519          DJNZ DCLCT1-$
084C 47       2520          LD   B,A
084D C9       2521          RET
              2522  ; SUBROUTINE TO UPDATE COORDINATES TO POINT AT NEXT CHARA
              2523  ; FRAME:
              2524  ; INPUT:        COORDINATES TAKEN FROM CBD,CBE IN CONTEXT
              2525  ; OUTPUT:       UPDATED COORDINATES RETURNED IN D AND E
              2526  ;               A,B = CLOBBERED, C=ENLARGE FACTOR CONVERT
084E CD3E08   2527  NXTFRM: CALL DCLCTB        ; GET ITERATION COUNT
0851 48       2528          LD   C,B           ; SAVE
0852 FD5605   2529          LD   D,(IY+CBD)    ; GET Y COORD
0855 FD7E04   2530          LD   A,(IY+CBE)    ; GET X COORD
0858 DD8601   2531  NXTFR1: ADD  A,(IX+FTFSX) ; ADD X FRAME SIZE
085B 10FB     2532          DJNZ NXTFR1-$      ; 2**ENLARGE TIMES
085D FEA0     2533          CP   160           ; PAST RIGHT EDGE OF SCREEN?
085F 3809     2534          JR   C,NXTFR3-$
0861 7A       2535          LD   A,D
0862 41       2536          LD   B,C
0863 DD8602   2537  NXTFR2: ADD  A,(IX+FTFSY) ; YEP - ADVANCE VERTICAL
0866 10FB     2538          DJNZ NXTFR2-$
0868 57       2539          LD   D,A
0869 AF       2540          XOR  A
086A 5F       2541  NXTFR3: LD   E,A
086B C9       2542          RET
              2543  ; SUBROUTINE TO WRITE ONE LINE OF A PATTERN WITH ENLARGE
              2544  ; AND EXPAND
              2545  ; ENTRY: HL = SOURCE   IX = FONT TABLE
086C DD4E03   2546  WRTLIN: LD   C,(IX+FTBYTE)
086F 0600     2547          LD   B,0
0871 DDE5     2548          PUSH IX            ; CAPTURE STACK POINTER
0873 DD210000 2549          LD   IX,0
0877 DD39     2550          ADD  IX,SP
0879 DDE5     2551          PUSH IX            ; SAVE CAPTURED STACK
087B D1       2552          POP  DE            ; DE = CAPTURED STACK
087C 3E0C     2553          LD   A,0CH         ; SET EXPAND TO 00,11
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

087E D319      2554           OUT  (XPAND),A
0880 3E08      2555           LD   A,08H           ; SET EXPAND BIT
0882 D30C      2556           OUT  (MAGIC),A
0884 FD7E06    2557           LD   A,(IY+CBC)      ; GET CONTROL BYTE
0887 E6C0      2558           AND  0C0H            ; ISOLATE ENLARGE AMOUNT
0889 2821      2559           JR   Z,WRTL3-$       ; JUMP IF ZERO
088B 07        2560           RLCA
088C 07        2561           RLCA
088D EB        2562  WRTL1:   EX   DE,HL
088E A7        2563           AND  A               ; CLEAR CARRY BIT
088F ED42      2564           SBC  HL,BC           ; COMPUTE STACK FRAME SIZE
0891 ED42      2565           SBC  HL,BC
0893 F9        2566           LD   SP,HL           ; SEIZE STACK SPACE
0894 CBB4      2567           RES  6,H             ; MAGICIFY THE ADDRESS
0896 F5        2568           PUSH AF
0897 41        2569           LD   B,C
0898 1A        2570  WRTL2:   LD   A,(DE)          ; GET SOURCE BYTE
0899 13        2571           INC  DE
089A 77        2572           LD   (HL),A          ; EXPAND IT
089B 23        2573           INC  HL
089C 77        2574           LD   (HL),A          ;
089D 23        2575           INC  HL
089E 10F8      2576           DJNZ WRTL2-$
08A0 CB21      2577           SLA  C
08A2 F1        2578           POP  AF
08A3 210000    2579           LD   HL,0            ; CAPTURE STACK TOP AGAIN
08A6 39        2580           ADD  HL,SP
08A7 54        2581           LD   D,H             ; SET DE=HL
08A8 5D        2582           LD   E,L             ; FOR NEXT DEST COMBO
08A9 3D        2583           DEC  A
08AA 20E1      2584           JR   NZ,WRTL1-$
               2585  ; NOW DO WRITE TO SCREEN
08AC CD3E08    2586  WRTL3:   CALL DCLCTB          ; GET ITERATION COUNTER
08AF CD7400    2587           CALL DELOAD
08B2 FD7E06    2588           LD   A,(IY+CBC)
08B5 D319      2589           OUT  (XPAND),A
08B7 E630      2590           AND  030H
08B9 F608      2591           OR   8
08BB CD080B    2592           CALL RELTA
08BE EB        2593           EX   DE,HL
08BF F5        2594  WRTL4:   PUSH AF
08C0 C5        2595           PUSH BC
08C1 D5        2596           PUSH DE
08C2 E5        2597           PUSH HL
08C3 41        2598           LD   B,C
08C4 1A        2599  WRTL5:   LD   A,(DE)
08C5 13        2600           INC  DE
08C6 77        2601           LD   (HL),A
08C7 23        2602           INC  HL
08C8 77        2603           LD   (HL),A
08C9 23        2604           INC  HL
08CA 10F8      2605           DJNZ WRTL5-$
08CC FD7E04    2606           LD   A,(IY+CBE)      ;
08CF E603      2607           AND  03
08D1 2801      2608           JR   Z,WRTL6-$       ;
08D3 70        2609           LD   (HL),B
08D4 E1        2610  WRTL6:   POP  HL              ; STEP TO NEXT LINE
```

```
08D5 0E28     2611          LD   C,BYTEPL
08D7 09       2612          ADD  HL,BC
08D8 D1       2613          POP  DE
08D9 C1       2614          POP  BC
08DA F1       2615          POP  AF
08DB D30C     2616          OUT  (MAGIC),A
08DD 10E0     2617          DJNZ WRTL4-$
08DF DDF9     2618          LD   SP,IX        ;  RESTORE STACK
08E1 DDE1     2619          POP  IX
08E3 C9       2620          RET

              2622    ; MACRO TO GENERATE CHARACTER PATTERN TABLE ENTRY
              2623  DEFCHR  MACR #A,#B,#C,#D,#E,#F,#G
              2624          DEFB #A
              2625          DEFB #B
              2626          DEFB #C
              2627          DEFB #D
              2628          DEFB #E
              2629          DEFB #F
              2630          DEFB #G
              2631          ENDM

              2633   ; LARGE CHARACTER SET (8 X 8)
08E4          2634  LRGCHR
08E4          2635          DEFCHR 000H,000H,000H,000H,000H,000H,000H ;  SPACE
08E4 00       2635 +        DEFB 000H
08E5 00       2635 +        DEFB 000H
08E6 00       2635 +        DEFB 000H
08E7 00       2635 +        DEFB 000H
08E8 00       2635 +        DEFB 000H
08E9 00       2635 +        DEFB 000H
08EA 00       2635 +        DEFB 000H
08EB          2636          DEFCHR 020H,020H,020H,020H,020H,000H,020H ;  !
08EB 20       2636 +        DEFB 020H
08EC 20       2636 +        DEFB 020H
08ED 20       2636 +        DEFB 020H
08EE 20       2636 +        DEFB 020H
08EF 20       2636 +        DEFB 020H
08F0 00       2636 +        DEFB 000H
08F1 20       2636 +        DEFB 020H
08F2          2637          DEFCHR 050H,050H,050H,000H,000H,000H,000H ;  "
08F2 50       2637 +        DEFB 050H
08F3 50       2637 +        DEFB 050H
08F4 50       2637 +        DEFB 050H
08F5 00       2637 +        DEFB 000H
08F6 00       2637 +        DEFB 000H
08F7 00       2637 +        DEFB 000H
08F8 00       2637 +        DEFB 000H
08F9          2638          DEFCHR 048H,048H,0FCH,048H,0FCH,048H,048H ;  #
08F9 48       2638 +        DEFB 048H
08FA 48       2638 +        DEFB 048H
08FB FC       2638 +        DEFB 0FCH
08FC 48       2638 +        DEFB 048H
```

```
ADDR OBJECT     STMT  LABEL    OPCD OPERAND        COMMENT

08FD FC         2638 +         DEFB 0FCH
08FE 48         2638 +         DEFB 048H
08FF 48         2638 +         DEFB 048H
0900            2639           DEFCHR 020H,078H,080H,070H,008H,0F0H,020H ;  $
0900 20         2639 +         DEFB 020H
0901 78         2639 +         DEFB 078H
0902 80         2639 +         DEFB 080H
0903 70         2639 +         DEFB 070H
0904 08         2639 +         DEFB 008H
0905 F0         2639 +         DEFB 0F0H
0906 20         2639 +         DEFB 020H
0907            2640           DEFCHR 0C0H,0C8H,010H,020H,040H,098H,018H ;  %
0907 C0         2640 +         DEFB 0C0H
0908 C8         2640 +         DEFB 0C8H
0909 10         2640 +         DEFB 010H
090A 20         2640 +         DEFB 020H
090B 40         2640 +         DEFB 040H
090C 98         2640 +         DEFB 098H
090D 18         2640 +         DEFB 018H
090E            2641           DEFCHR 060H,090H,0A0H,040H,0A8H,090H,068H ;  &
090E 60         2641 +         DEFB 060H
090F 90         2641 +         DEFB 090H
0910 A0         2641 +         DEFB 0A0H
0911 40         2641 +         DEFB 040H
0912 A8         2641 +         DEFB 0A8H
0913 90         2641 +         DEFB 090H
0914 68         2641 +         DEFB 068H
0915            2642           DEFCHR 060H,060H,060H,000H,000H,000H,000H ;  '
0915 60         2642 +         DEFB 060H
0916 60         2642 +         DEFB 060H
0917 60         2642 +         DEFB 060H
0918 00         2642 +         DEFB 000H
0919 00         2642 +         DEFB 000H
091A 00         2642 +         DEFB 000H
091B 00         2642 +         DEFB 000H
091C            2643           DEFCHR 010H,020H,020H,020H,020H,020H,010H ;  (
091C 10         2643 +         DEFB 010H
091D 20         2643 +         DEFB 020H
091E 20         2643 +         DEFB 020H
091F 20         2643 +         DEFB 020H
0920 20         2643 +         DEFB 020H
0921 20         2643 +         DEFB 020H
0922 10         2643 +         DEFB 010H
0923            2644           DEFCHR 040H,020H,020H,020H,020H,020H,040H ;  )
0923 40         2644 +         DEFB 040H
0924 20         2644 +         DEFB 020H
0925 20         2644 +         DEFB 020H
0926 20         2644 +         DEFB 020H
0927 20         2644 +         DEFB 020H
0928 20         2644 +         DEFB 020H
0929 40         2644 +         DEFB 040H
092A            2645           DEFCHR 000H,0A8H,070H,0D8H,070H,0A8H,000H ;  *
092A 00         2645 +         DEFB 000H
092B A8         2645 +         DEFB 0A8H
092C 70         2645 +         DEFB 070H
092D D8         2645 +         DEFB 0D8H
092E 70         2645 +         DEFB 070H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

092F A8        2645 +         DEFB 0A8H
0930 00        2645 +         DEFB 000H
0931           2646           DEFCHR 000H,020H,020H,0F8H,020H,020H,000H ;  +
0931 00        2646 +         DEFB 000H
0932 20        2646 +         DEFB 020H
0933 20        2646 +         DEFB 020H
0934 F8        2646 +         DEFB 0F8H
0935 20        2646 +         DEFB 020H
0936 20        2646 +         DEFB 020H
0937 00        2646 +         DEFB 000H
0938           2647           DEFCHR 000H,000H,000H,060H,060H,020H,040H ;  ,
0938 00        2647 +         DEFB 000H
0939 00        2647 +         DEFB 000H
093A 00        2647 +         DEFB 000H
093B 60        2647 +         DEFB 060H
093C 60        2647 +         DEFB 060H
093D 20        2647 +         DEFB 020H
093E 40        2647 +         DEFB 040H
093F           2648           DEFCHR 000H,000H,000H,0F8H,000H,000H,000H ;  -
093F 00        2648 +         DEFB 000H
0940 00        2648 +         DEFB 000H
0941 00        2648 +         DEFB 000H
0942 F8        2648 +         DEFB 0F8H
0943 00        2648 +         DEFB 000H
0944 00        2648 +         DEFB 000H
0945 00        2648 +         DEFB 000H
0946           2649           DEFCHR 000H,000H,000H,000H,000H,060H,060H ;  .
0946 00        2649 +         DEFB 000H
0947 00        2649 +         DEFB 000H
0948 00        2649 +         DEFB 000H
0949 00        2649 +         DEFB 000H
094A 00        2649 +         DEFB 000H
094B 60        2649 +         DEFB 060H
094C 60        2649 +         DEFB 060H
094D           2650           DEFCHR 000H,008H,010H,020H,040H,080H,000H ;
094D 00        2650 +         DEFB 000H
094E 08        2650 +         DEFB 008H
094F 10        2650 +         DEFB 010H
0950 20        2650 +         DEFB 020H
0951 40        2650 +         DEFB 040H
0952 80        2650 +         DEFB 080H
0953 00        2650 +         DEFB 000H
0954           2651           DEFCHR 070H,088H,088H,088H,088H,088H,070H ;  0
0954 70        2651 +         DEFB 070H
0955 88        2651 +         DEFB 088H
0956 88        2651 +         DEFB 088H
0957 88        2651 +         DEFB 088H
0958 88        2651 +         DEFB 088H
0959 88        2651 +         DEFB 088H
095A 70        2651 +         DEFB 070H
095B           2652           DEFCHR 020H,060H,020H,020H,020H,020H,070H ;  1
095B 20        2652 +         DEFB 020H
095C 60        2652 +         DEFB 060H
095D 20        2652 +         DEFB 020H
095E 20        2652 +         DEFB 020H
095F 20        2652 +         DEFB 020H
0960 20        2652 +         DEFB 020H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

0961 70        2652 +         DEFB 070H
0962           2653           DEFCHR 070H,088H,008H,070H,080H,080H,0F8H ; 2
0962 70        2653 +         DEFB 070H
0963 88        2653 +         DEFB 088H
0964 08        2653 +         DEFB 008H
0965 70        2653 +         DEFB 070H
0966 80        2653 +         DEFB 080H
0967 80        2653 +         DEFB 080H
0968 F8        2653 +         DEFB 0F8H
0969           2654           DEFCHR 070H,088H,008H,030H,008H,088H,070H ; 3
0969 70        2654 +         DEFB 070H
096A 88        2654 +         DEFB 088H
096B 08        2654 +         DEFB 008H
096C 30        2654 +         DEFB 030H
096D 08        2654 +         DEFB 008H
096E 88        2654 +         DEFB 088H
096F 70        2654 +         DEFB 070H
0970           2655           DEFCHR 010H,030H,050H,090H,0F8H,010H,010H ; 4
0970 10        2655 +         DEFB 010H
0971 30        2655 +         DEFB 030H
0972 50        2655 +         DEFB 050H
0973 90        2655 +         DEFB 090H
0974 F8        2655 +         DEFB 0F8H
0975 10        2655 +         DEFB 010H
0976 10        2655 +         DEFB 010H
0977           2656           DEFCHR 0F8H,080H,0F0H,008H,008H,088H,070H ; 5
0977 F8        2656 +         DEFB 0F8H
0978 80        2656 +         DEFB 080H
0979 F0        2656 +         DEFB 0F0H
097A 08        2656 +         DEFB 008H
097B 08        2656 +         DEFB 008H
097C 88        2656 +         DEFB 088H
097D 70        2656 +         DEFB 070H
097E           2657           DEFCHR 030H,040H,080H,0F0H,088H,088H,070H ; 6
097E 30        2657 +         DEFB 030H
097F 40        2657 +         DEFB 040H
0980 80        2657 +         DEFB 080H
0981 F0        2657 +         DEFB 0F0H
0982 88        2657 +         DEFB 088H
0983 88        2657 +         DEFB 088H
0984 70        2657 +         DEFB 070H
0985           2658           DEFCHR 0F8H,008H,010H,020H,040H,040H,040H ; 7
0985 F8        2658 +         DEFB 0F8H
0986 08        2658 +         DEFB 008H
0987 10        2658 +         DEFB 010H
0988 20        2658 +         DEFB 020H
0989 40        2658 +         DEFB 040H
098A 40        2658 +         DEFB 040H
098B 40        2658 +         DEFB 040H
098C           2659           DEFCHR 070H,088H,088H,070H,088H,088H,070H ; 8
098C 70        2659 +         DEFB 070H
098D 88        2659 +         DEFB 088H
098E 88        2659 +         DEFB 088H
098F 70        2659 +         DEFB 070H
0990 88        2659 +         DEFB 088H
0991 88        2659 +         DEFB 088H
0992 70        2659 +         DEFB 070H
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

0993           2660          DEFCHR 070H,088H,088H,078H,008H,010H,060H ;  9
0993 70        2660 +        DEFB 070H
0994 88        2660 +        DEFB 088H
0995 88        2660 +        DEFB 088H
0996 78        2660 +        DEFB 078H
0997 08        2660 +        DEFB 008H
0998 10        2660 +        DEFB 010H
0999 60        2660 +        DEFB 060H
099A           2661          DEFCHR 000H,060H,060H,000H,060H,060H,000H ;  :
099A 00        2661 +        DEFB 000H
099B 60        2661 +        DEFB 060H
099C 60        2661 +        DEFB 060H
099D 00        2661 +        DEFB 000H
099E 60        2661 +        DEFB 060H
099F 60        2661 +        DEFB 060H
09A0 00        2661 +        DEFB 000H
09A1           2662          DEFCHR 060H,060H,000H,060H,060H,020H,040H ;  ;
09A1 60        2662 +        DEFB 060H
09A2 60        2662 +        DEFB 060H
09A3 00        2662 +        DEFB 000H
09A4 60        2662 +        DEFB 060H
09A5 60        2662 +        DEFB 060H
09A6 20        2662 +        DEFB 020H
09A7 40        2662 +        DEFB 040H
09A8           2663          DEFCHR 010H,020H,040H,080H,040H,020H,010H ;  <
09A8 10        2663 +        DEFB 010H
09A9 20        2663 +        DEFB 020H
09AA 40        2663 +        DEFB 040H
09AB 80        2663 +        DEFB 080H
09AC 40        2663 +        DEFB 040H
09AD 20        2663 +        DEFB 020H
09AE 10        2663 +        DEFB 010H
09AF           2664          DEFCHR 000H,000H,0F8H,000H,0F8H,000H,000H ;  =
09AF 00        2664 +        DEFB 000H
09B0 00        2664 +        DEFB 000H
09B1 F8        2664 +        DEFB 0F8H
09B2 00        2664 +        DEFB 000H
09B3 F8        2664 +        DEFB 0F8H
09B4 00        2664 +        DEFB 000H
09B5 00        2664 +        DEFB 000H
09B6           2665          DEFCHR 040H,020H,010H,008H,010H,020H,040H ;  >
09B6 40        2665 +        DEFB 040H
09B7 20        2665 +        DEFB 020H
09B8 10        2665 +        DEFB 010H
09B9 08        2665 +        DEFB 008H
09BA 10        2665 +        DEFB 010H
09BB 20        2665 +        DEFB 020H
09BC 40        2665 +        DEFB 040H
09BD           2666          DEFCHR 070H,088H,008H,010H,020H,000H,020H ;  ?
09BD 70        2666 +        DEFB 070H
09BE 88        2666 +        DEFB 088H
09BF 08        2666 +        DEFB 008H
09C0 10        2666 +        DEFB 010H
09C1 20        2666 +        DEFB 020H
09C2 00        2666 +        DEFB 000H
09C3 20        2666 +        DEFB 020H
09C4           2667          DEFCHR 070H,088H,0B8H,0A8H,0B8H,080H,078H ;  @
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

09C4 70       2667 +        DEFB 070H
09C5 88       2667 +        DEFB 088H
09C6 B8       2667 +        DEFB 0B8H
09C7 A8       2667 +        DEFB 0A8H
09C8 B8       2667 +        DEFB 0B8H
09C9 80       2667 +        DEFB 080H
09CA 78       2667 +        DEFB 078H
09CB          2668          DEFCHR 070H,088H,088H,0F8H,088H,088H,088H ;  A
09CB 70       2668 +        DEFB 070H
09CC 88       2668 +        DEFB 088H
09CD 88       2668 +        DEFB 088H
09CE F8       2668 +        DEFB 0F8H
09CF 88       2668 +        DEFB 088H
09D0 88       2668 +        DEFB 088H
09D1 88       2668 +        DEFB 088H
09D2          2669          DEFCHR 0F0H,088H,088H,0F0H,088H,088H,0F0H ;  B
09D2 F0       2669 +        DEFB 0F0H
09D3 88       2669 +        DEFB 088H
09D4 88       2669 +        DEFB 088H
09D5 F0       2669 +        DEFB 0F0H
09D6 88       2669 +        DEFB 088H
09D7 88       2669 +        DEFB 088H
09D8 F0       2669 +        DEFB 0F0H
09D9          2670          DEFCHR 070H,088H,080H,080H,080H,088H,070H ;  C
09D9 70       2670 +        DEFB 070H
09DA 88       2670 +        DEFB 088H
09DB 80       2670 +        DEFB 080H
09DC 80       2670 +        DEFB 080H
09DD 80       2670 +        DEFB 080H
09DE 88       2670 +        DEFB 088H
09DF 70       2670 +        DEFB 070H
09E0          2671          DEFCHR 0F0H,088H,088H,088H,088H,088H,0F0H ;  D
09E0 F0       2671 +        DEFB 0F0H
09E1 88       2671 +        DEFB 088H
09E2 88       2671 +        DEFB 088H
09E3 88       2671 +        DEFB 088H
09E4 88       2671 +        DEFB 088H
09E5 88       2671 +        DEFB 088H
09E6 F0       2671 +        DEFB 0F0H
09E7          2672          DEFCHR 0F8H,080H,080H,0E0H,080H,080H,0F8H ;  E
09E7 F8       2672 +        DEFB 0F8H
09E8 80       2672 +        DEFB 080H
09E9 80       2672 +        DEFB 080H
09EA E0       2672 +        DEFB 0E0H
09EB 80       2672 +        DEFB 080H
09EC 80       2672 +        DEFB 080H
09ED F8       2672 +        DEFB 0F8H
09EE          2673          DEFCHR 0F8H,080H,080H,0E0H,080H,080H,080H ;  F
09EE F8       2673 +        DEFB 0F8H
09EF 80       2673 +        DEFB 080H
09F0 80       2673 +        DEFB 080H
09F1 E0       2673 +        DEFB 0E0H
09F2 80       2673 +        DEFB 080H
09F3 80       2673 +        DEFB 080H
09F4 80       2673 +        DEFB 080H
09F5          2674          DEFCHR 070H,088H,080H,080H,098H,088H,078H ;  G
09F5 70       2674 +        DEFB 070H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

09F6 88        2674 +         DEFB 088H
09F7 80        2674 +         DEFB 080H
09F8 80        2674 +         DEFB 080H
09F9 98        2674 +         DEFB 098H
09FA 88        2674 +         DEFB 088H
09FB 78        2674 +         DEFB 078H
09FC           2675           DEFCHR 088H,088H,088H,0F8H,088H,088H,088H ;  H
09FC 88        2675 +         DEFB 088H
09FD 88        2675 +         DEFB 088H
09FE 88        2675 +         DEFB 088H
09FF F8        2675 +         DEFB 0F8H
0A00 88        2675 +         DEFB 088H
0A01 88        2675 +         DEFB 088H
0A02 88        2675 +         DEFB 088H
0A03           2676           DEFCHR 070H,020H,020H,020H,020H,020H,070H ;  I
0A03 70        2676 +         DEFB 070H
0A04 20        2676 +         DEFB 020H
0A05 20        2676 +         DEFB 020H
0A06 20        2676 +         DEFB 020H
0A07 20        2676 +         DEFB 020H
0A08 20        2676 +         DEFB 020H
0A09 70        2676 +         DEFB 070H
0A0A           2677           DEFCHR 008H,008H,008H,008H,008H,088H,070H ;  J
0A0A 08        2677 +         DEFB 008H
0A0B 08        2677 +         DEFB 008H
0A0C 08        2677 +         DEFB 008H
0A0D 08        2677 +         DEFB 008H
0A0E 08        2677 +         DEFB 008H
0A0F 88        2677 +         DEFB 088H
0A10 70        2677 +         DEFB 070H
0A11           2678           DEFCHR 088H,090H,0A0H,0C0H,0A0H,090H,088H ;  K
0A11 88        2678 +         DEFB 088H
0A12 90        2678 +         DEFB 090H
0A13 A0        2678 +         DEFB 0A0H
0A14 C0        2678 +         DEFB 0C0H
0A15 A0        2678 +         DEFB 0A0H
0A16 90        2678 +         DEFB 090H
0A17 88        2678 +         DEFB 088H
0A18           2679           DEFCHR 080H,080H,080H,080H,080H,080H,0F8H ;  L
0A18 80        2679 +         DEFB 080H
0A19 80        2679 +         DEFB 080H
0A1A 80        2679 +         DEFB 080H
0A1B 80        2679 +         DEFB 080H
0A1C 80        2679 +         DEFB 080H
0A1D 80        2679 +         DEFB 080H
0A1E F8        2679 +         DEFB 0F8H
0A1F           2680           DEFCHR 088H,0D8H,0A8H,0A8H,088H,088H,088H ;  M
0A1F 88        2680 +         DEFB 088H
0A20 D8        2680 +         DEFB 0D8H
0A21 A8        2680 +         DEFB 0A8H
0A22 A8        2680 +         DEFB 0A8H
0A23 88        2680 +         DEFB 088H
0A24 88        2680 +         DEFB 088H
0A25 88        2680 +         DEFB 088H
0A26           2681           DEFCHR 088H,0C8H,0A8H,098H,088H,088H,088H ;  N
0A26 88        2681 +         DEFB 088H
0A27 C8        2681 +         DEFB 0C8H
```

```
0A28 A8        2681 +         DEFB 0A8H
0A29 98        2681 +         DEFB 098H
0A2A 88        2681 +         DEFB 088H
0A2B 88        2681 +         DEFB 088H
0A2C 88        2681 +         DEFB 088H
0A2D           2682           DEFCHR 0F8H,088H,088H,088H,088H,088H,0F8H ; O
0A2D F8        2682 +         DEFB 0F8H
0A2E 88        2682 +         DEFB 088H
0A2F 88        2682 +         DEFB 088H
0A30 88        2682 +         DEFB 088H
0A31 88        2682 +         DEFB 088H
0A32 88        2682 +         DEFB 088H
0A33 F8        2682 +         DEFB 0F8H
0A34           2683           DEFCHR 0F0H,088H,088H,0F0H,080H,080H,080H ; P
0A34 F0        2683 +         DEFB 0F0H
0A35 88        2683 +         DEFB 088H
0A36 88        2683 +         DEFB 088H
0A37 F0        2683 +         DEFB 0F0H
0A38 80        2683 +         DEFB 080H
0A39 80        2683 +         DEFB 080H
0A3A 80        2683 +         DEFB 080H
0A3B           2684           DEFCHR 070H,088H,088H,088H,0A8H,090H,068H ; Q
0A3B 70        2684 +         DEFB 070H
0A3C 88        2684 +         DEFB 088H
0A3D 88        2684 +         DEFB 088H
0A3E 88        2684 +         DEFB 088H
0A3F A8        2684 +         DEFB 0A8H
0A40 90        2684 +         DEFB 090H
0A41 68        2684 +         DEFB 068H
0A42           2685           DEFCHR 0F0H,088H,088H,0F0H,0A0H,090H,088H ; R
0A42 F0        2685 +         DEFB 0F0H
0A43 88        2685 +         DEFB 088H
0A44 88        2685 +         DEFB 088H
0A45 F0        2685 +         DEFB 0F0H
0A46 A0        2685 +         DEFB 0A0H
0A47 90        2685 +         DEFB 090H
0A48 88        2685 +         DEFB 088H
0A49           2686           DEFCHR 070H,088H,080H,070H,008H,088H,070H ; S
0A49 70        2686 +         DEFB 070H
0A4A 88        2686 +         DEFB 088H
0A4B 80        2686 +         DEFB 080H
0A4C 70        2686 +         DEFB 070H
0A4D 08        2686 +         DEFB 008H
0A4E 88        2686 +         DEFB 088H
0A4F 70        2686 +         DEFB 070H
0A50           2687           DEFCHR 0F8H,020H,020H,020H,020H,020H,020H ; T
0A50 F8        2687 +         DEFB 0F8H
0A51 20        2687 +         DEFB 020H
0A52 20        2687 +         DEFB 020H
0A53 20        2687 +         DEFB 020H
0A54 20        2687 +         DEFB 020H
0A55 20        2687 +         DEFB 020H
0A56 20        2687 +         DEFB 020H
0A57           2688           DEFCHR 088H,088H,088H,088H,088H,088H,070H ; U
0A57 88        2688 +         DEFB 088H
0A58 88        2688 +         DEFB 088H
0A59 88        2688 +         DEFB 088H
```

```
0A5A 88        2688 +         DEFB 088H
0A5B 88        2688 +         DEFB 088H
0A5C 88        2688 +         DEFB 088H
0A5D 70        2688 +         DEFB 070H
0A5E           2689           DEFCHR 088H,088H,088H,050H,050H,020H,020H ;  V
0A5E 88        2689 +         DEFB 088H
0A5F 88        2689 +         DEFB 088H
0A60 88        2689 +         DEFB 088H
0A61 50        2689 +         DEFB 050H
0A62 50        2689 +         DEFB 050H
0A63 20        2689 +         DEFB 020H
0A64 20        2689 +         DEFB 020H
0A65           2690           DEFCHR 088H,088H,088H,0A8H,0A8H,0D8H,088H ;  W
0A65 88        2690 +         DEFB 088H
0A66 88        2690 +         DEFB 088H
0A67 88        2690 +         DEFB 088H
0A68 A8        2690 +         DEFB 0A8H
0A69 A8        2690 +         DEFB 0A8H
0A6A D8        2690 +         DEFB 0D8H
0A6B 88        2690 +         DEFB 088H
0A6C           2691           DEFCHR 088H,088H,050H,020H,050H,088H,088H ;  X
0A6C 88        2691 +         DEFB 088H
0A6D 88        2691 +         DEFB 088H
0A6E 50        2691 +         DEFB 050H
0A6F 20        2691 +         DEFB 020H
0A70 50        2691 +         DEFB 050H
0A71 88        2691 +         DEFB 088H
0A72 88        2691 +         DEFB 088H
0A73           2692           DEFCHR 088H,088H,050H,020H,020H,020H,020H ;  Y
0A73 88        2692 +         DEFB 088H
0A74 88        2692 +         DEFB 088H
0A75 50        2692 +         DEFB 050H
0A76 20        2692 +         DEFB 020H
0A77 20        2692 +         DEFB 020H
0A78 20        2692 +         DEFB 020H
0A79 20        2692 +         DEFB 020H
0A7A           2693           DEFCHR 0F8H,008H,010H,020H,040H,080H,0F8H ;  Z
0A7A F8        2693 +         DEFB 0F8H
0A7B 08        2693 +         DEFB 008H
0A7C 10        2693 +         DEFB 010H
0A7D 20        2693 +         DEFB 020H
0A7E 40        2693 +         DEFB 040H
0A7F 80        2693 +         DEFB 080H
0A80 F8        2693 +         DEFB 0F8H
0A81           2694           DEFCHR 070H,040H,040H,040H,040H,040H,070H ;  [
0A81 70        2694 +         DEFB 070H
0A82 40        2694 +         DEFB 040H
0A83 40        2694 +         DEFB 040H
0A84 40        2694 +         DEFB 040H
0A85 40        2694 +         DEFB 040H
0A86 40        2694 +         DEFB 040H
0A87 70        2694 +         DEFB 070H
0A88           2695           DEFCHR 000H,080H,040H,020H,010H,008H,000H ;  \
0A88 00        2695 +         DEFB 000H
0A89 80        2695 +         DEFB 080H
0A8A 40        2695 +         DEFB 040H
0A8B 20        2695 +         DEFB 020H
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

0A8C 10        2695 +        DEFB 010H
0A8D 08        2695 +        DEFB 008H
0A8E 00        2695 +        DEFB 000H
0A8F           2696          DEFCHR 070H,010H,010H,010H,010H,010H,070H ;  ]
0A8F 70        2696 +        DEFB 070H
0A90 10        2696 +        DEFB 010H
0A91 10        2696 +        DEFB 010H
0A92 10        2696 +        DEFB 010H
0A93 10        2696 +        DEFB 010H
0A94 10        2696 +        DEFB 010H
0A95 70        2696 +        DEFB 070H
0A96           2697          DEFCHR 020H,070H,0A8H,020H,020H,020H,020H ;  ^
0A96 20        2697 +        DEFB 020H
0A97 70        2697 +        DEFB 070H
0A98 A8        2697 +        DEFB 0A8H
0A99 20        2697 +        DEFB 020H
0A9A 20        2697 +        DEFB 020H
0A9B 20        2697 +        DEFB 020H
0A9C 20        2697 +        DEFB 020H
0A9D           2698          DEFCHR 000H,020H,040H,0F8H,040H,020H,000H ;  ←
0A9D 00        2698 +        DEFB 000H
0A9E 20        2698 +        DEFB 020H
0A9F 40        2698 +        DEFB 040H
0AA0 F8        2698 +        DEFB 0F8H
0AA1 40        2698 +        DEFB 040H
0AA2 20        2698 +        DEFB 020H
0AA3 00        2698 +        DEFB 000H
0AA4           2699          DEFCHR 020H,020H,020H,020H,0A8H,070H,020H ;  DOWN
0AA4 20        2699 +        DEFB 020H
0AA5 20        2699 +        DEFB 020H
0AA6 20        2699 +        DEFB 020H
0AA7 20        2699 +        DEFB 020H
0AA8 A8        2699 +        DEFB 0A8H
0AA9 70        2699 +        DEFB 070H
0AAA 20        2699 +        DEFB 020H
0AAB           2700          DEFCHR 000H,020H,010H,0F8H,010H,020H,000H ;  RIGHT
0AAB 00        2700 +        DEFB 000H
0AAC 20        2700 +        DEFB 020H
0AAD 10        2700 +        DEFB 010H
0AAE F8        2700 +        DEFB 0F8H
0AAF 10        2700 +        DEFB 010H
0AB0 20        2700 +        DEFB 020H
0AB1 00        2700 +        DEFB 000H
0AB2           2701          DEFCHR 000H,088H,050H,020H,050H,088H,000H ;  MULTI
0AB2 00        2701 +        DEFB 000H
0AB3 88        2701 +        DEFB 088H
0AB4 50        2701 +        DEFB 050H
0AB5 20        2701 +        DEFB 020H
0AB6 50        2701 +        DEFB 050H
0AB7 88        2701 +        DEFB 088H
0AB8 00        2701 +        DEFB 000H
0AB9 00        2702          DEFB 0
0ABA 20        2703          DEFB 20H
0ABB 00        2704          DEFB 0
0ABC F8        2705          DEFB 0F8H
0ABD 00        2706          DEFB 0
0ABE 20        2707          DEFB 20H
```

```
               2708  ; ** LAST BYTE OF DIVIDE IS ZERO, WHICH HAPPENS TO BE FIR
               2709  ;     BYTE OF ...
               2710   ; SMALL CHARACTERS (4 X 6)
0ABF           2711  SMLCHR
0ABF           2712           DEF5 000H,000H,000H,000H,000H ;  SPACE
0ABF 00        2712 +         DEFB 000H
0AC0 00        2712 +         DEFB 000H
0AC1 00        2712 +         DEFB 000H
0AC2 00        2712 +         DEFB 000H
0AC3 00        2712 +         DEFB 000H


0AC4 DDE1      2714  MMJUMP:  POP  IX
0AC6 E3        2715           EX   (SP),HL
0AC7 DDE9      2716           JP   (IX)


               2718  ;NAME:   CONVERT KEY CODE TO ASCII
               2719  ;PURPOSE:   SAME
               2720  ;INPUT:   A=KEY CODE
               2721  ;OUTPUT:   A=ASCII EQUIVALENT
               2722  ;HOW:   TABLE LOOKUP
0AC9           2723  MKCTAS:
0AC9 48        2724           LD   C,B
0ACA 0600      2725           LD   B,0
0ACC 21D50A    2726           LD   HL,KCTATB
0ACF 09        2727           ADD  HL,BC
0AD0 7E        2728           LD   A,(HL)
0AD1 FD7709    2729  QFROG:   LD   (IY+CBA),A
0AD4 C9        2730           RET

0AD5           2732  KCTATB:
0AD5 20        2733           DEFB ' '            ;SPACE
0AD6 43        2734           DEFB 'C'            ;BULLET
0AD7 5E        2735           DEFB 5EH            ;UP ARROW
0AD8 5C        2736           DEFB 5CH            ;DOWN ARROW
0AD9 25        2737           DEFB '%'            ;
0ADA 52        2738           DEFB 'R'            ;RECALL
0ADB 53        2739           DEFB 'S'            ;STORE
0ADC 3B        2740           DEFB ';'            ;PLUS-MINUS
0ADD 2F        2741           DEFB '/'            ; DIVIDE
0ADE 37        2742           DEFB '7'
0ADF 38        2743           DEFB '8'
0AE0 39        2744           DEFB '9'
0AE1 2A        2745           DEFB '*'            ;TIMES
0AE2 34        2746           DEFB '4'
0AE3 35        2747           DEFB '5'
0AE4 36        2748           DEFB '6'
0AE5 2D        2749           DEFB '-'            ;MINUS
0AE6 31        2750           DEFB '1'
0AE7 32        2751           DEFB '2'
0AE8 33        2752           DEFB '3'
0AE9 2B        2753           DEFB '+'            ;PLUS
0AEA 26        2754           DEFB '&'            ;CE
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

0AEB 30        2755           DEFB '0'
0AEC 2E        2756           DEFB '.'            ;POINT
0AED 3D        2757           DEFB '='            ;EQUALS


               2759   ;  NAME:           FILL AREA
               2760   ;  PURPOSE:        SET REGION OF SCREEN TO CONSTANT VALUE
               2761   ;  INPUT:          A = DATA TO FILL WITH
               2762   ;                  BC = NUMBER OF BYTES TO FILL
               2763   ;                  DE = STARTING ADDRESS OF REGION TO FILL
0AEE EB        2764  MFILL:   EX   DE,HL
0AEF 77        2765  MFILL1:  LD   (HL),A         ; STUFF BYTE
0AF0 EDA1      2766           CPI                 ; BUMP HL, DEC BC
0AF2 EAEF0A    2767           JP   PE,MFILL1
0AF5 C9        2768           RET


               2770   ;  NAME:           RELATIVE TO ABSOLUTE
               2771   ;  PURPOSE:        COORDINATE CONVERSION
               2772   ;  INPUT:          E = X COORDINATE
               2773   ;                  D = Y COORDINATE
               2774   ;                  A = MAGIC REGISTER VALUE TO USE
               2775   ;  OUTPUT:         DE = ABSOLUTE ADDRESS
               2776   ;                  A = MAGIC REGISTER TO USE
               2777  ; MAGIC ENTRY POINT
0AF6 CD080B    2778  MRELAB:  CALL RELTA
0AF9 1805      2779           JR   MRELA2-$
               2780  ; NONMAGIC ENTRY POINT
0AFB CD4E0B    2781  MRELA1:  CALL RELTA1
0AFE CBF2      2782           SET  6,D            ; NONMAGIC THE ADDRESS
0B00 FD7304    2783  MRELA2:  LD   (IY+CBE),E     ; UPDATE CB DE
0B03 FD7205    2784           LD   (IY+CBD),D
0B06 18C9      2785  MFROG:   JR   QFROG-$
               2786  ; MAGIC ENTRY POINT
0B08 CD4E0B    2787  RELTA:   CALL RELTA1
0B0B D30C      2788           OUT  (MAGIC),A
0B0D C9        2789           RET
0B0E 00        2790  CKSUM2:  DEFB 0              ; *** CHECKSUM ***
0B0F           2791           DEF5 0E0H,0A0H,0A0H,0A0H,0E0H ;  0
0B0F E0        2791 +         DEFB 0E0H
0B10 A0        2791 +         DEFB 0A0H
0B11 A0        2791 +         DEFB 0A0H
0B12 A0        2791 +         DEFB 0A0H
0B13 E0        2791 +         DEFB 0E0H
0B14           2792           DEF5 040H,040H,040H,040H,040H ;  1
0B14 40        2792 +         DEFB 040H
0B15 40        2792 +         DEFB 040H
0B16 40        2792 +         DEFB 040H
0B17 40        2792 +         DEFB 040H
0B18 40        2792 +         DEFB 040H
0B19           2793           DEF5 0E0H,020H,0E0H,080H,0E0H ;  2
0B19 E0        2793 +         DEFB 0E0H
0B1A 20        2793 +         DEFB 020H
0B1B E0        2793 +         DEFB 0E0H
0B1C 80        2793 +         DEFB 080H
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND      COMMENT

0B1D E0       2793 +         DEFB 0E0H
0B1E          2794           DEF5 0E0H, 020H, 060H, 020H, 0E0H ;  3
0B1E E0       2794 +         DEFB 0E0H
0B1F 20       2794 +         DEFB 020H
0B20 60       2794 +         DEFB 060H
0B21 20       2794 +         DEFB 020H
0B22 E0       2794 +         DEFB 0E0H
0B23          2795           DEF5 0A0H, 0A0H, 0E0H, 020H, 020H ;  4
0B23 A0       2795 +         DEFB 0A0H
0B24 A0       2795 +         DEFB 0A0H
0B25 E0       2795 +         DEFB 0E0H
0B26 20       2795 +         DEFB 020H
0B27 20       2795 +         DEFB 020H
0B28          2796           DEF5 0E0H, 080H, 0E0H, 020H, 0E0H ;  5
0B28 E0       2796 +         DEFB 0E0H
0B29 80       2796 +         DEFB 080H
0B2A E0       2796 +         DEFB 0E0H
0B2B 20       2796 +         DEFB 020H
0B2C E0       2796 +         DEFB 0E0H
0B2D          2797           DEF5 0E0H, 080H, 0E0H, 0A0H, 0E0H ;  6
0B2D E0       2797 +         DEFB 0E0H
0B2E 80       2797 +         DEFB 080H
0B2F E0       2797 +         DEFB 0E0H
0B30 A0       2797 +         DEFB 0A0H
0B31 E0       2797 +         DEFB 0E0H
0B32          2798           DEF5 0E0H, 020H, 020H, 020H, 020H ;  7
0B32 E0       2798 +         DEFB 0E0H
0B33 20       2798 +         DEFB 020H
0B34 20       2798 +         DEFB 020H
0B35 20       2798 +         DEFB 020H
0B36 20       2798 +         DEFB 020H
0B37          2799           DEF5 0E0H, 0A0H, 0E0H, 0A0H, 0E0H ;  8
0B37 E0       2799 +         DEFB 0E0H
0B38 A0       2799 +         DEFB 0A0H
0B39 E0       2799 +         DEFB 0E0H
0B3A A0       2799 +         DEFB 0A0H
0B3B E0       2799 +         DEFB 0E0H
0B3C          2800           DEF5 0E0H, 0A0H, 0E0H, 020H, 0E0H ;  9
0B3C E0       2800 +         DEFB 0E0H
0B3D A0       2800 +         DEFB 0A0H
0B3E E0       2800 +         DEFB 0E0H
0B3F 20       2800 +         DEFB 020H
0B40 E0       2800 +         DEFB 0E0H
0B41          2801           DEF5 000H, 040H, 000H, 040H, 000H ;  :
0B41 00       2801 +         DEFB 000H
0B42 40       2801 +         DEFB 040H
0B43 00       2801 +         DEFB 000H
0B44 40       2801 +         DEFB 040H
0B45 00       2801 +         DEFB 000H
0B46          2802           DEF5 040H, 0E0H, 0E0H, 0E0H, 0E0H ;  BULLET
0B46 40       2802 +         DEFB 040H
0B47 E0       2802 +         DEFB 0E0H
0B48 E0       2802 +         DEFB 0E0H
0B49 E0       2802 +         DEFB 0E0H
0B4A E0       2802 +         DEFB 0E0H
```

```
              2804  ; MOVE ROUTINE
0B4B EDB0     2805  MMOVE:  LDIR
0B4D C9       2806          RET


              2808   ; SYSTEM ENTRY POINT FOR NONMAGIC ADDRESSES
0B4E E5       2809  RELTA1: PUSH HL
0B4F E6FC     2810          AND  0FCH           ; TOSS OUT SHIFT AMOUNT
0B51 6F       2811          LD   L,A            ; SAVE
0B52 7B       2812          LD   A,E            ; GET X
0B53 E603     2813          AND  03H            ; ISOLATE SHIFT AMOUNT
0B55 B5       2814          OR   L              ; COMBINE WITH MR
0B56 F5       2815  RELTA2: PUSH AF
0B57 E640     2816          AND  040H           ; IS FLOPPED BIT SET?
0B59 7B       2817          LD   A,E
0B5A 2803     2818          JR   Z,RELTA3-$     ; JUMP IF NOT
0B5C 2F       2819          CPL                 ; YEP - UNFLOP THE COORDINATE
0B5D C6A0     2820          ADD  A,160
0B5F 6A       2821  RELTA3: LD   L,D            ; HL = Y
0B60 2600     2822          LD   H,0
0B62 29       2823          ADD  HL,HL          ; SET HL = Y * 8
0B63 29       2824          ADD  HL,HL
0B64 29       2825          ADD  HL,HL
0B65 54       2826          LD   D,H
0B66 5D       2827          LD   E,L
0B67 29       2828          ADD  HL,HL          ; SET HL = Y * 32
0B68 29       2829          ADD  HL,HL
0B69 19       2830          ADD  HL,DE          ; SET HL = Y * 40
0B6A CB3F     2831          SRL  A              ; A = X 4
0B6C CB3F     2832          SRL  A
0B6E 5F       2833          LD   E,A
0B6F 1600     2834          LD   D,0
0B71 19       2835          ADD  HL,DE          ; HL = Y * 40 + X          4
              2836          IF   NWHDWR-1
              2837          ENDIF
0B72 EB       2838          EX   DE,HL


              2840  ; NAME:             RETURN FROM MACRO SUBROUTINE
              2841  ; PURPOSE:          RETURN CONTROL TO CALLER
              2842  ; THIS CODE WAS 'STOLEN' FROM RELABS SINCE
              2843  ; IT DOES THE STACK CLEANUP THAT MRET DOES
0B73 F1       2844  MMRET:  POP  AF
0B74 E1       2845          POP  HL
0B75 C9       2846          RET


              2848  ; ENTRY FOR USER
0B76 CD7B0B   2849  INXNIB: CALL XNIB
0B79 188B     2850          JR   MFROG-$
```

```
               2852  ; NAME:          INDEX NIBBLE
               2853  ; PURPOSE:       LOAD OF SPECIFIED NIBBLE RELATIVE TO BASE
               2854  ; INPUT:         C = NIBBLE NUMBER
               2855  ;                HL = BASE ADDRESS
               2856  ; OUTPUT:        NIBBLE RETURNED RIGHT JUSTIFIED IN A.
               2857  ; DESCRIPTION:   BYTE = NIBBLE#         2+BASE
               2858  ;  THE LOW ORDER NIBBLE OF A GIVEN BYTE IS ADDRESSED
               2859  ; BY AN EVEN NIBBLE NUMBER.
0B7B E5        2860  XNIB:   PUSH HL
0B7C C5        2861          PUSH BC
0B7D 0600      2862          LD   B,0
0B7F CB39      2863          SRL  C
0B81 09        2864          ADD  HL,BC
0B82 7E        2865          LD   A,(HL)
0B83 C1        2866          POP  BC
0B84 CB41      2867          BIT  0,C
0B86 2804      2868          JR   Z,XNIB1-$
0B88 0F        2869          RRCA
0B89 0F        2870          RRCA
0B8A 0F        2871          RRCA
0B8B 0F        2872          RRCA
0B8C E60F      2873  XNIB1:  AND  0FH
0B8E E1        2874          POP  HL
0B8F C9        2875          RET


               2877  ; NAME:          STORE NIBBLE
               2878  ; PURPOSE:       NIBBLE STORING (!)
               2879  ; INPUT:         A = NIBBLE TO STORE
               2880  ;                C = NIBBLE NUMBER (AS IN XNIB)
               2881  ;       HL = BASE ADDRESS
0B90 E5        2882  PUTNIB: PUSH HL
0B91 C5        2883          PUSH BC
0B92 0600      2884          LD   B,0
0B94 CB39      2885          SRL  C
0B96 09        2886          ADD  HL,BC
0B97 C1        2887          POP  BC
0B98 CB41      2888          BIT  0,C
0B9A 2809      2889          JR   Z,PUTNB1-$
               2890  ; H.O. CASE - SHIFT IT
0B9C 07        2891          RLCA
0B9D 07        2892          RLCA
0B9E 07        2893          RLCA
0B9F 07        2894          RLCA
0BA0 AE        2895          XOR  (HL)
0BA1 E6F0      2896          AND  0F0H
0BA3 1803      2897          JR   PUTNB2-$
0BA5 AE        2898  PUTNB1: XOR  (HL)           ; L.O.  CASE
0BA6 E60F      2899          AND  0FH
0BA8 AE        2900  PUTNB2: XOR  (HL)
0BA9 77        2901          LD   (HL),A
0BAA E1        2902          POP  HL
0BAB C9        2903          RET
```

```
               2905  ; NAME :  INDEX WORD TABLE (WORD INDEX)
               2906  ; PURPOSE: TO INDEX AN ARRAY OF DEFW'S
               2907  ; INPUTS: A=INDEX NUMBER (0-255)
               2908  ; HL -> TABLE ENTRY 0
               2909  ; OUTPUTS:         DE = ENTRY LOOKED UP
               2910  ;                  HL = POINTER TO ENTRY IN TABLE
0BAC 5F        2911  MINDW:   LD   E,A
0BAD 1600      2912           LD   D,0
0BAF CB23      2913           SLA  E
0BB1 CB12      2914           RL   D               ; DE*2
0BB3 19        2915           ADD  HL,DE
0BB4 5E        2916           LD   E,(HL)
0BB5 23        2917           INC  HL
0BB6 56        2918           LD   D,(HL)
0BB7 2B        2919           DEC  HL
0BB8 CDF40C    2920  STHLDE:  CALL FINDL3
0BBB 1808      2921           JR   MINDB1-$        ; JOIN STORE IN INDEX BYTE

               2923  ; NAME:            INDEX BYTE TABLE
               2924  ; PURPOSE:         TABLE LOOKUP
               2925  ; INPUTS:          A = INDEX NUMBER
               2926  ; OUTPUT:          A = VALUE OF BYTE
               2927  ;                  HL = POINTER TO TABLE ENTRY
0BBD 5F        2928  MINDB:   LD   E,A
0BBE 1600      2929           LD   D,0
0BC0 19        2930           ADD  HL,DE
0BC1 7E        2931           LD   A,(HL)
0BC2 FD7709    2932           LD   (IY+CBA),A
0BC5 FD740B    2933  MINDB1:  LD   (IY+CBH),H
0BC8 FD750A    2934           LD   (IY+CBL),L
0BCB C9        2935           RET

               2937  ; NAME:     DISPLAY TIME
               2938  ; PURPOSE:  DISPLAY TIME ON SCREEN
               2939  ; INPUTS:   E = X COORD
               2940  ;           D = Y COORD
               2941  ;           C = SAME AS DISCHR OPTIONS EXCEPT BIT 7 = 1
               2942  ;               TO DISPLAY COLON AND SECONDS
               2943  ; OUTPUTS:  NONE
0BCC           2944  MDISTI:
0BCC DD210D02  2945           LD   IX,SMLFNT
0BD0 0642      2946           LD   B,42H
0BD2 21EE4F    2947           LD   HL,GTMINS
0BD5 C5        2948           PUSH BC
0BD6 FDCB06BE  2949           RES  7,(IY+CBC)
0BDA CDEB0B    2950           CALL BCDISP
0BDD C1        2951           POP  BC
0BDE CB79      2952           BIT  7,C
0BE0 C8        2953           RET  Z
0BE1 3EBA      2954           LD   A,80H+3AH
```

```
0BE3 CDE107   2955          CALL DISPCH
0BE6 0642     2956          LD   B,42H
0BE8 21ED4F   2957          LD   HL,GTSECS
              2958  ; AND FALL INTO ...


              2960  ; NAME:           DISPLAY BCD NUMBER
              2961  ; INPUT:          B = NUMBER DISPLAY OPTIONS
              2962  ;                 C = CHARACTER DISPLAY OPTIONS
              2963  ;                 DE = Y,X COORDINATES
              2964  ;                 HL = NUMBER ADDRESS (POINTS AT LO BYTE)
              2965  ;                 IX = ALTERNATE FONT (IF USED)
              2966  ; OUTPUT:         DE UPDATED
              2967  ; DESCRIPTION:  THIS ROUTINE CONVERTS EACH NIBBLE INTO
              2968  ; ASCII AND DISPLAYS IT.  THE NORMALLY ILLEGAL BCD
              2969  ; VALUES ARE DISPLAYED AS CODES 2A THRU 2F RESPECTIVELY.
              2970  ; THE NUMBER DISPLAY OPTIONS BYTE IS FORMATED AS FOLLOWS:
              2971  ; BIT 7           SET IF LEADING ZERO SURPRESSION WANTED
              2972  ; BIT 6           SET IF USE OF ALTERNATE FONT WANTED
              2973  ; BITS 5-0        NUMBER OF DIGITS TO DISPLAY (NOT NUMBER O
0BEB 78       2974  BCDISP: LD   A,B           ; GET OPTIONS
0BEC E63F     2975          AND  3FH           ; ISOLATE NUMBER OF DIGITS
0BEE 3D       2976  BCDD0:  DEC  A
0BEF F8       2977          RET  M             ; QUIT IF NULL OR NO MORE
0BF0 4F       2978          LD   C,A           ; SAVE
0BF1 CD7B0B   2979          CALL XNIB          ; GET NEXT DIGIT
0BF4 2007     2980          JR   NZ,BCDD1-$    ; JUMP IF NONZERO
0BF6 CB78     2981          BIT  7,B           ; IS ZERO SURPRESS ON?
0BF8 2803     2982          JR   Z,BCDD1-$     ; JUMP IF NOT
0BFA B1       2983          OR   C             ; LAST DIGIT?
0BFB 2014     2984          JR   NZ,BCDD4-$    ; JUMP IF NOT
0BFD CBB8     2985  BCDD1:  RES  7,B           ; CLEAR LEADING ZERO FLAG
0BFF C606     2986          ADD  A,6
0C01 E60F     2987          AND  0FH
0C03 C62A     2988          ADD  A,2AH
0C05 CB70     2989  BCDD2:  BIT  6,B           ; ALTERNATE FONT?
0C07 2802     2990          JR   Z,BCDD3-$     ; JUMP IF NO
0C09 F680     2991          OR   80H           ; YEA - SET THE BIT
0C0B CDE107   2992  BCDD3:  CALL DISPCH        ; DISPLAY THE CHAR
0C0E 79       2993          LD   A,C           ; GET LOOP COUNTER IN A
0C0F 18DD     2994          JR   BCDD0-$       ; AND GO FOR NEXT
0C11 3E20     2995  BCDD4:  LD   A,' '         ; LEADING ZERO - WRITE A SPACE
0C13 18F0     2996          JR   BCDD2-$


              2998  ; NAME:     INCREMENT SCORE
              2999  ; PURPOSE:  INCREMENT SCORE AND COMPARE TO END SCORE
              3000  ; INPUTS:   HL -> PLAYER SCORE LOW ADDR OF 3 BYTES
              3001  ; OUTPUTS:  GSBEND OF GAMSTB SET IF MAX SCORE REACHED
0C15 0603     3002  MINCSC: LD   B,3
0C17 E5       3003          PUSH HL
0C18 7E       3004  INCLOP: LD   A,(HL)
0C19 C601     3005          ADD  A,1
0C1B 27       3006          DAA
0C1C 77       3007          LD   (HL),A
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

0C1D 2003      3008           JR   NZ,CMPIT-$
0C1F 23        3009           INC  HL
0C20 10F6      3010           DJNZ INCLOP-$
0C22 E1        3011  CMPIT:   POP  HL
0C23 23        3012           INC  HL
0C24 23        3013           INC  HL
0C25 3AF84F    3014           LD   A,(GAMSTB)
0C28 CB4F      3015           BIT  GSBSCR,A
0C2A C8        3016           RET  Z
0C2B 11F64F    3017           LD   DE,ENDSCR+2
0C2E 0603      3018           LD   B,3
0C30 1A        3019  CMPLOP:  LD   A,(DE)
0C31 BE        3020           CP   (HL)
0C32 2807      3021           JR   Z,REPEAT-$     ;ENDSCR = SCORE
0C34 D0        3022           RET  NC             ;ENDSCR > SCORE
0C35 21F84F    3023  SETEND:  LD   HL,GAMSTB      ;ENDSCR < SCORE
0C38 CBFE      3024           SET  GSBEND,(HL)
0C3A C9        3025           RET
0C3B 1B        3026  REPEAT:  DEC  DE
0C3C 2B        3027           DEC  HL
0C3D 10F1      3028           DJNZ CMPLOP-$
0C3F 18F4      3029           JR   SETEND-$


               3031  ; NAME:           QUIT
               3032  ; PURPOSE:        HOLD PRESENT GAME SCORE UNTIL KEY HIT OR
               3033  ; SAY GAME OVER
0C41           3034  MQUIT:   SYSSUK STRDIS
0C41 FF        3034 +         RST  56
0C42 35        3034 +         DEFB STRDIS+1
               3034 +         IF   STRDIS.EQ.INTPC
               3034 +         ENDIF
0C43 30        3035           DEFB 48
0C44 18        3036           DEFB 24
0C45 4C        3037           DEFB 01001100B
0C46 570C      3038           DEFW GMOVR
0C48           3039           SYSTEM ACTINT       ; ACTIVATE INTERRUPTS
0C48 FF        3039 +         RST  56
0C49 0E        3039 +         DEFB ACTINT
               3039 +         IF   ACTINT.EQ.INTPC
               3039 +         ENDIF
0C4A           3040  MQUIT1:  SYSSUK SENTRY       ; WAIT FOR SOMETHING TO HAPPEN
0C4A FF        3040 +         RST  56
0C4B 43        3040 +         DEFB SENTRY+1
               3040 +         IF   SENTRY.EQ.INTPC
               3040 +         ENDIF
0C4C 1402      3041           DEFW AKEYS
0C4E FE14      3042           CP   STO
0C50 2804      3043           JR   Z,MQUIT2-$     ; TRIGGER CHANGE?
0C52 FE13      3044           CP   SKYD           ; KEY HIT?
0C54 20F4      3045           JR   NZ,MQUIT1-$    ; NO - KEEP GOING
0C56 C7        3046  MQUIT2:  RST  0              ; YES - RESET
0C57 47414D45  3047  GMOVR:   DEFM 'GAME'
0C5B 06        3048           DEFB 6
0C5C 4F564552  3049           DEFM 'OVER'
0C60 00        3050           DEFB 0
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT



                3052  ; *****************
                3053  ; * MENU ROUTINES *
                3054  ; *****************
>0060           3055  NOLINE   EQU  96            ; NUMBER OF DISPLAYED LINES
>0000           3056  MNNL     EQU  0             ; NEXT FIELD
>0001           3057  MNNH     EQU  1
>0002           3058  MNSAL    EQU  2             ; STRING ADDRESS
>0003           3059  MNSAH    EQU  3
>0004           3060  MNGL     EQU  4             ; GO TO ADDRESS
>0005           3061  MNGH     EQU  5


                3063  ; SYSTEM POWER UP ROUTINE
 0C61 3A0020    3064  PWRUP:   LD   A,(FIRSTC)    ; GET FIRST CASSETTE LOCATION
 0C64 FEC3      3065           CP   0C3H          ; IS IT A JUMP??
 0C66 CA0020    3066           JP   Z,FIRSTC      ; JUMP TO IT IF SO
 0C69 31CE4F    3067           LD   SP,BEGRAM
 0C6C           3068           SYSSUK FILL        ; CLEAR SYSTEM RAM
 0C6C FF        3068 +         RST  56
 0C6D 1B        3068 +         DEFB FILL+1
                3068 +         IF   FILL.EQ.INTPC
                3068 +         ENDIF
 0C6E CE4F      3069           DEFW BEGRAM
 0C70 3200      3070           DEFW 50
 0C72 00        3071           DEFB 0
 0C73 32FF0F    3072           LD   (WASTE),A     ; CLEAR SHIFTER
 0C76 3D        3073           DEC  A
 0C77 32EC4F    3074           LD   (TIMOUT),A    ; CLEAR TIMEOUT WATCHDOG
 0C7A           3075           SYSTEM INTPC
 0C7A FF        3075 +         RST  56
 0C7B 00        3075 +         DEFB INTPC
                3075 +         IF   INTPC.EQ.INTPC
>0001           3075 +INTP@    DEFL 1
                3075 +         ENDIF
 0C7C           3076           DO   EMUSIC
 0C7C 15        3076 +         DEFB EMUSIC+1
 0C7D           3077           DO   SETOUT
 0C7D 17        3077 +         DEFB SETOUT+1
 0C7E BF        3078           DEFB (NOLINE*2)-1
 0C7F 29        3079           DEFB 41
 0C80 08        3080           DEFB 8
 0C81           3081           DO   COLSET
 0C81 19        3081 +         DEFB COLSET+1
 0C82 1300      3082           DEFW MENUCL
 0C84           3083           DO   ACTINT
 0C84 0F        3083 +         DEFB ACTINT+1
 0C85           3084           EXIT
 0C85 02        3084 +         DEFB XINTC
>0000           3084 +INTP@    DEFL 0
 0C86 11F30D    3085           LD   DE,GAMSTR     ; 'SELECT GAME' AS TITLE
 0C89 210020    3086           LD   HL,FIRSTC     ; ASSUME MENU STARTS IN CASSETT
```

```
ADDR OBJECT    STMT   LABEL    OPCD OPERAND        COMMENT

0C8C 7E        3087            LD   A,(HL)         ; GET FIRST CASSETTE BYTE
0C8D 23        3088            INC  HL
0C8E FE55      3089            CP   55H            ; IS SENTINEL THERE?
0C90 2803      3090            JR   Z,PWRUP1-$     ; YEP - JUMP
0C92 211802    3091            LD   HL,GUNLNK      ; WRONG - USE ONBOARD ONLY
0C95           3092   PWRUP1:  SYSTEM MENU         ; DISPLAY THE MENU
0C95 FF        3092 +          RST  56
0C96 4A        3092 +          DEFB MENU
               3092 +          IF   MENU.EQ.INTPC
               3092 +          ENDIF


               3094   ; NAME:           DISPLAY MENU AND BRANCH ON CHOICE
               3095   ; INPUT:          HL = MENU LIST
               3096   ;                 DE = MENU TITLE
               3097   ; OUTPUT:         DE = TITLE OF SELECTION MADE
               3098   ; DESCRIPTION:
               3099   ;        THE MENU LIST IS A LINKED LIST OF THE FOLLOWING F
               3100   ; *************************
               3101   ; * 0 * NEXT ENTRY        *
               3102   ; * 1 *                   *
               3103   ; *************************
               3104   ; * 2 * STRING ADDRESS    *
               3105   ; * 3 *                   *
               3106   ; *************************
               3107   ; * 4 * BRANCH TO ADDRESS *
               3108   ; * 5 *                   *
               3109   ; *************************
               3110   ; THIS LIST IS TERMINATED BY A NEXT ENTRY FIELD OF ZEROS
               3111   ; A MAXIMUM OF EIGHT ENTRYS MAY BE DISPLAYED.
0C97 E5        3112   MMENU:   PUSH HL
0C98 E5        3113            PUSH HL
0C99 CD190D    3114            CALL MNCLR          ; CLEAR SCREEN AND THROWUP TITL
0C9C           3115            XYRELL DE,16,12
0C9C 11100C    3115 +          LD   DE,.RES.(12).SHL.8+(16)
0C9F 010901    3116            LD   BC,109H        ; INITIALIZE ENTRY # AND COLOR
0CA2 DDE1      3117   MMENU1:  POP  IX             ; FIRST ENTRY TO IX
0CA4 78        3118            LD   A,B            ; SELECTION NUMBER TO A
0CA5 C630      3119            ADD  A,'0'          ; MAKE IT ASCII
0CA7           3120            SYSTEM CHRDIS       ; AND SHOW IT
0CA7 FF        3120 +          RST  56
0CA8 32        3120 +          DEFB CHRDIS
               3120 +          IF   CHRDIS.EQ.INTPC
               3120 +          ENDIF
0CA9 3E2D      3121            LD   A,'-'          ; DISPLAY DASH
0CAB           3122            SYSTEM CHRDIS
0CAB FF        3122 +          RST  56
0CAC 32        3122 +          DEFB CHRDIS
               3122 +          IF   CHRDIS.EQ.INTPC
               3122 +          ENDIF
0CAD DD6603    3123            LD   H,(IX+MNSAH)   ; HL = STRING ADDRESS
0CB0 DD6E02    3124            LD   L,(IX+MNSAL)
0CB3           3125            SYSTEM STRDIS       ; DISPLAY SELECTION
0CB3 FF        3125 +          RST  56
0CB4 34        3125 +          DEFB STRDIS
               3125 +          IF   STRDIS.EQ.INTPC
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

              3125 +        ENDIF
0CB5 3E08     3126          LD   A,8
0CB7 82       3127          ADD  A,D          ; TO NEXT LINE
0CB8 57       3128          LD   D,A
0CB9 1E10     3129          LD   E,16
0CBB 04       3130          INC  B            ; BUMP ENTRY #
0CBC DD6601   3131          LD   H,(IX+MNNH)  ; HL = NEXT ENTRY ADDR
0CBF DD6E00   3132          LD   L,(IX+MNNL)
0CC2 E5       3133          PUSH HL
0CC3 7C       3134          LD   A,H
0CC4 B5       3135          OR   L
0CC5 20DB     3136          JR   NZ,MMENU1-$  ; NO - JUMP BACK
              3137  ; AT THIS POINT HL = 0, (SP) = 0
0CC7 39       3138          ADD  HL,SP        ; HL = STACK POINTER
0CC8 C5       3139  MMENU3: PUSH BC
0CC9 010101   3140          LD   BC,0101H
0CCC          3141          XYRELL DE,16,77   ; FEEDBACK ADDRESS
0CCC 11104D   3141 +        LD   DE,.RES.(77).SHL.8+(16)
0CCF          3142          SYSTEM GETNUM     ; GET NUMBA
0CCF FF       3142 +        RST  56
0CD0 4E       3142 +        DEFB GETNUM
              3142 +        IF   GETNUM.EQ.INTPC
              3142 +        ENDIF
0CD1 C1       3143          POP  BC
0CD2 7E       3144          LD   A,(HL)       ; HOW DOES SHE LOOK?
0CD3 A7       3145          AND  A            ; ZERO ENTERED?
0CD4 2803     3146          JR   Z,MMENU5-$   ; JUMP IF SO
0CD6 B8       3147          CP   B            ; IN RANGE?
0CD7 3806     3148          JR   C,MMENU6-$   ; JUMP IF SO
0CD9 3E3F     3149  MMENU5: LD   A,'?'        ; DUD ENTRY - SHOW ?
0CDB          3150          SYSTEM CHRDIS
0CDB FF       3150 +        RST  56
0CDC 32       3150 +        DEFB CHRDIS
              3150 +        IF   CHRDIS.EQ.INTPC
              3150 +        ENDIF
0CDD 18E9     3151          JR   MMENU3-$     ; GO BACK FOR NEXT TRY
0CDF E1       3152  MMENU6: POP  HL           ; THROW OUT ENTRY AREA
0CE0 D1       3153          POP  DE           ; RESTORE HEAD OF MENU LIST
0CE1 47       3154          LD   B,A          ; NUMBER ENTERED TO B
0CE2 EB       3155  MMENU7: EX   DE,HL        ; HL = ENTRY PTR
0CE3 5E       3156          LD   E,(HL)       ; DE = NEXT
0CE4 23       3157          INC  HL
0CE5 56       3158          LD   D,(HL)
0CE6 10FA     3159          DJNZ MMENU7-$     ; COUNT DOWN TO ENTRY
0CE8 23       3160          INC  HL
0CE9 5E       3161          LD   E,(HL)       ; STRING TO DE
0CEA 23       3162          INC  HL
0CEB 56       3163          LD   D,(HL)
0CEC 23       3164          INC  HL
0CED 4E       3165          LD   C,(HL)       ; GO TO ADDRESS TO BC
0CEE 23       3166          INC  HL
0CEF 46       3167          LD   B,(HL)
0CF0 E1       3168          POP  HL           ; HL = RETURN TO PLACE
0CF1 F1       3169          POP  AF           ; THROW OUT OLD PC
0CF2 C5       3170          PUSH BC           ; PUT NEW PC ON STACK
0CF3 E5       3171          PUSH HL           ; AND PUT BACK DUMMY RETURN
0CF4 FD7304   3172  FINDL3: LD   (IY+CBE),E   ; PASS BACK TITLE ADDRESS
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

0CF7 FD7205   3173          LD   (IY+CBD),D
0CFA C9       3174          RET                ; AND GO BACK


              3176  ; NAME:          GET PARAMETER
              3177  ; PURPOSE:       INPUT OF PROGRAM OPTIONS
              3178  ; INPUT:         A = NUMBER OF DIGITS
              3179  ;                BC = PROMPT STRING ADDRESS
              3180  ;                DE = FRAME TITLE ADDRESS
              3181  ;                HL = PARAMETER ADDRESS
              3182  ; DESCRIPTION:
              3183  ;       THIS ROUTINE ASKS THE USER TO ENTER A NUMBER
              3184  ; FIRST A MENU FRAME IS CREATED, USING THE STRING
              3185  ; POINTED AT BY DE AS A TITLE.  THE STRING 'ENTER'
              3186  ; IS DISPLAYED, FOLLOWED BY THE PROMPT STRING.
              3187  ; GETNUM IS THEN CALLED TO INPUT THE NUMBER.  FEEDBACK
              3188  ; IS PROVIDED IN DOUBLE SIZED CHARACTERS.
              3189  ; NOTE: ** THIS ROUTINE USES TWO SYSTEM LEVELS AND THE AL
0CFB F5       3190  MGETP:  PUSH AF            ; SAVE NUMBER OF DIGITS
0CFC E5       3191          PUSH HL
0CFD C5       3192          PUSH BC
0CFE CD190D   3193          CALL MNCLR
0D01          3194          SYSSUK STRDIS      ; DISPLAY 'ENTER'
0D01 FF       3194 +        RST  56
0D02 35       3194 +        DEFB STRDIS+1
              3194 +        IF   STRDIS.EQ.INTPC
              3194 +        ENDIF
0D03 08       3195          DEFB 8
0D04 20       3196          DEFB 32
0D05 09       3197          DEFB 1001B
0D06 B70D     3198          DEFW ENTSTG
0D08 E1       3199          POP  HL
0D09          3200          SYSTEM STRDIS      ; DISPLAY WHAT TO ENTER
0D09 FF       3200 +        RST  56
0D0A 34       3200 +        DEFB STRDIS
              3200 +        IF   STRDIS.EQ.INTPC
              3200 +        ENDIF
0D0B E1       3201          POP  HL
0D0C F1       3202          POP  AF
0D0D 47       3203          LD   B,A
0D0E CBF1     3204          SET  6,C           ; SET LARGE CHARS
0D10          3205          XYRELL DE,48,48    ; LOAD FEEDBACK ADDRESS
0D10 113030   3205 +        LD   DE,.RES.(48).SHL.8+(48)
0D13          3206          SYSTEM GETNUM      ; GET NUMBER
0D13 FF       3206 +        RST  56
0D14 4E       3206 +        DEFB GETNUM
              3206 +        IF   GETNUM.EQ.INTPC
              3206 +        ENDIF
0D15          3207          SYSSUK PAWS        ; LET USER READ IT
0D15 FF       3207 +        RST  56
0D16 51       3207 +        DEFB PAWS+1
              3207 +        IF   PAWS.EQ.INTPC
              3207 +        ENDIF
0D17 0F       3208          DEFB 15
0D18 C9       3209          RET
              3210  ; SUBROUTINE TO CLEAR SCREEN FOR MENU AND THROWUP TITLE
```

```
0D19 D5        3211   MNCLR:  PUSH DE
0D1A           3212           SYSSUK FILL
0D1A FF        3212 +         RST  56
0D1B 1B        3212 +         DEFB FILL+1
               3212 +         IF   FILL.EQ.INTPC
               3212 +         ENDIF
0D1C 0040      3213           DEFW NORMEM
0D1E B801      3214           DEFW 11*BYTEPL
0D20 00        3215           DEFB 0
0D21           3216           SYSSUK FILL
0D21 FF        3216 +         RST  56
0D22 1B        3216 +         DEFB FILL+1
               3216 +         IF   FILL.EQ.INTPC
               3216 +         ENDIF
0D23 B841      3217           DEFW NORMEM+(11*BYTEPL)
0D25 480D      3218           DEFW (NOLINE-11)*BYTEPL
0D27 55        3219           DEFB 55H
0D28 E1        3220           POP  HL
0D29           3221           XYRELL DE,24,0    ; TITLE
0D29 111800    3221 +         LD   DE,.RES.(0).SHL.8+(24)
0D2C 0E04      3222           LD   C,0100B
0D2E           3223           SYSTEM STRDIS
0D2E FF        3223 +         RST  56
0D2F 34        3223 +         DEFB STRDIS
               3223 +         IF   STRDIS.EQ.INTPC
               3223 +         ENDIF
0D30 C9        3224           RET


               3226  ; NAME:          GET NUMBER
               3227  ; INPUT:         B = DISNUM OPTIONS
               3228  ;                C = CHRDIS OPTIONS FOR FEEDBACK
               3229  ;                DE = COORDINATES OF FEEDBACK AREA
               3230  ;                HL = ADDRESS OF WHERE TO STASH NUMBER
               3231  ; DESCRIPTION:   THIS ROUTINE CAN INPUT A NUMBER FROM
               3232  ;        EITHER THE KEYBOARD OR THE HAND CONTROL.  KEYBOAR
               3233  ;        ENTRY PROCEEDS CONVENTIONALY.  GETNUM EXITS
               3234  ;        WHEN THE EQUALS KEY IS PRESSED OR THE REQUIRED NU
               3235  ;        OF DIGITS IS ENTERED
               3236  ;                PLAYER ONE HAND CONTROL MAY ALSO BE USED
               3237  ;        ENTER A NUMBER.  TO USE THIS OPTION, PULL THE TRI
               3238  ;        THEN ROTATE THE POT UNTIL THE NUMBER YOU WISH TO
               3239  ;        ENTER IS SHOWN IN THE FEEDBACK AREA.  PULL THE TR
               3240  ;        AGAIN TO REGISTER THE ENTRY.  IF DURING THIS PROC
               3241  ;        THE KEYBOARD IS USED - KEYBOARD INPUT WILL OVERRI
0D31 D9        3242  MGETN:   EXX
0D32 CD990D    3243           CALL CLRNUM       ; CLEAR THE NUMBER
0D35 4F        3244           LD   C,A          ; SET ZERO DIGITS IN - POT ENAB
0D36 FD7E07    3245  MGETN1:  LD   A,(IY+CBB)   ; ENTRY COMPLETE?
0D39 A9        3246           XOR  C
0D3A E63F      3247           AND  3FH
0D3C C8        3248           RET  Z            ; QUIT IF SO
0D3D 21360D    3249           LD   HL,MGETN1
0D40 E5        3250           PUSH HL
0D41           3251           SYSTEM RANGED     ; RANDOMIZE WHILE WE WAIT
0D41 FF        3251 +         RST  56
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

0D42 76       3251 +        DEFB RANGED
              3251 +        IF   RANGED.EQ.INTPC
              3251 +        ENDIF
0D43          3252          SYSSUK SENTRY
0D43 FF       3252 +        RST  56
0D44 43       3252 +        DEFB SENTRY+1
              3252 +        IF   SENTRY.EQ.INTPC
              3252 +        ENDIF
0D45 0B00     3253          DEFW NUMBAS
0D47          3254          SYSSUK DOIT
0D47 FF       3254 +        RST  56
0D48 45       3254 +        DEFB DOIT+1
              3254 +        IF   DOIT.EQ.INTPC
              3254 +        ENDIF
0D49 4C0D     3255          DEFW GNUMDO
0D4B C9       3256          RET               ; NOTHIN - LOOP ON SENTRY
0D4C          3257  GNUMDO: JMP  SKYD,MGETN6
0D4C 13       3257 +        DEFB SKYD
0D4D 7F0D     3257 +        DEFW MGETN6
              3257 +        IF   0
              3257 +        ENDIF
0D4F          3258          JMP  STO,MGETN2
0D4F 14       3258 +        DEFB STO
0D50 550D     3258 +        DEFW MGETN2
              3258 +        IF   0
              3258 +        ENDIF
0D52          3259          JMP  SPO,MGETN3
0D52 1C       3259 +        DEFB SPO
0D53 610D     3259 +        DEFW MGETN3
              3259 +        IF   0
              3259 +        ENDIF
              3260  ; TRIGGER ROUTINE
0D55 CB60     3261  MGETN2: BIT  4,B          ; 0-1 TRANS?
0D57 C8       3262          RET  Z            ; NO - IGNORE
0D58 79       3263          LD   A,C
0D59 3C       3264          INC  A            ; ARE WE ALREADY IN POT MODE?
0D5A 283A     3265          JR   Z,MGETN9-$   ; YEP - JUMP TO EXIT
0D5C CB79     3266          BIT  7,C          ; POT LEGAL?
0D5E C0       3267          RET  NZ           ; NO - IGNORE
0D5F 0EFF     3268          LD   C,0FFH       ; SET POT FLAG
              3269  ; POT ROUTINE
0D61 79       3270  MGETN3: LD   A,C          ; QUIT IF NOT IN POT MODE
0D62 3C       3271          INC  A
0D63 C0       3272          RET  NZ
              3273  ; HOW MANY DIGITS?
0D64 D9       3274          EXX               ; TO NORMAL SET
0D65 78       3275          LD   A,B          ; SNATCH DIGITS
0D66 D9       3276          EXX
0D67 FE01     3277          CP   1            ; 1 PRAY TELL?
0D69 060A     3278          LD   B,10
0D6B 2802     3279          JR   Z,MGETN4-$   ; JUMP IF GOOD GUESS
0D6D 0664     3280          LD   B,100        ; WRONG!
0D6F DB1C     3281  MGETN4: IN   A,(POTO)     ; GET CURRENT POT VALUE
0D71 57       3282          LD   D,A          ; RANGE IT
0D72 AF       3283          XOR  A
0D73 5F       3284          LD   E,A
0D74 67       3285          LD   H,A
```

```
0D75 19         3286  MGETN5: ADD  HL,DE
0D76 CE00       3287          ADC  A,0             ; ADD EVERY CARRY TO AC
0D78 27         3288          DAA
0D79 10FA       3289          DJNZ MGETN5-$
0D7B D9         3290          EXX                  ; BACK TO NORMAL SET
0D7C 77         3291          LD   (HL),A
0D7D 1814       3292          JR   MGETN8-$
                3293  ; KEYBOARD ROUTINE
0D7F 0C         3294  MGETN6: INC  C               ; POT MODE?
0D80 2004       3295          JR   NZ,MGETN7-$     ; JUMP IF NOT
0D82 CD990D     3296          CALL CLRNUM
0D85 0C         3297          INC  C               ; SET ONE DIGIT SO FAR
0D86 CBF9       3298  MGETN7: SET  7,C             ; SET POT LOCKOUT
0D88            3299          SYSTEM KCTASC
0D88 FF         3299 +        RST  56
0D89 40         3299 +        DEFB KCTASC
                3299 +        IF   KCTASC.EQ.INTPC
                3299 +        ENDIF
0D8A FE3D       3300          CP   '='             ; EQUALS TYPED?
0D8C 2808       3301          JR   Z,MGETN9-$      ; QUIT IF EQUALS
0D8E E60F       3302          AND  0FH
0D90 D9         3303          EXX
0D91            3304          SYSTEM SHIFTU        ; SHIFT DIGIT UP
0D91 FF         3304 +        RST  56
0D92 60         3304 +        DEFB SHIFTU
                3304 +        IF   SHIFTU.EQ.INTPC
                3304 +        ENDIF
0D93 D5         3305  MGETN8: PUSH DE
0D94            3306          SYSTEM DISNUM
0D94 FF         3306 +        RST  56
0D95 36         3306 +        DEFB DISNUM
                3306 +        IF   DISNUM.EQ.INTPC
                3306 +        ENDIF
                3307  ; ENTER HERE FOR EQUAL OR TRIGGER EXIT TO THROW OUT RETUR
0D96 D1         3308  MGETN9: POP  DE
0D97 D9         3309          EXX                  ; BACK TO NORMAL
0D98 C9         3310          RET


                3312  ; SUBROUTINE TO CLEAR NUMBER
0D99 C5         3313  CLRNUM: PUSH BC
0D9A D9         3314          EXX                  ; TO NORMAL SET
0D9B E5         3315          PUSH HL
0D9C 78         3316          LD   A,B
0D9D 3C         3317          INC  A
0D9E E63E       3318          AND  3EH
0DA0 1F         3319          RRA                  ;
0DA1 D9         3320          EXX                  ; BACK TO ALTERNATE SET
0DA2 4F         3321          LD   C,A
0DA3 AF         3322          XOR  A
0DA4 47         3323          LD   B,A
0DA5 D1         3324          POP  DE
0DA6            3325          SYSTEM FILL
0DA6 FF         3325 +        RST  56
0DA7 1A         3325 +        DEFB FILL
                3325 +        IF   FILL.EQ.INTPC
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

              3325 +        ENDIF
0DA8 C1       3326          POP  BC
0DA9 C9       3327          RET


              3329  ; NAME:           SHIFT UP
              3330  ; INPUT:          A = DATA TO SHIFT UP
              3331  ;                 B = SIZE IN DIGITS
              3332  ;                 HL = AREA TO SHIFT ADDRESS
0DAA F5       3333  MSHFTU: PUSH AF
0DAB 78       3334          LD   A,B
0DAC 3C       3335          INC  A
0DAD E63E     3336          AND  3EH
0DAF 47       3337          LD   B,A
0DB0 F1       3338          POP  AF
0DB1 ED6F     3339  SHFTU1: RLD
0DB3 23       3340          INC  HL
0DB4 10FB     3341          DJNZ SHFTU1-$
0DB6 C9       3342          RET



0DB7 454E5445 3344  ENTSTG: DEFM 'ENTER '
0DBD 00       3345          DEFB 0
0DBE FA01     3346  CML:    DEFW CALCL
0DC0 D30D     3347          DEFW PNCM
0DC2 2813     3348          DEFW CMSTRT       ; CHECKMATE START
0DC4 0000     3349  SCBL:   DEFW 0
0DC6 E80D     3350          DEFW PNSCB
0DC8 190E     3351          DEFW SCBST
0DCA 47554E46 3352  PNGF:   DEFM 'GUNFIGHT'
0DD2 00       3353          DEFB 0
0DD3 43484543 3354  PNCM:   DEFM 'CHECKMATE'
0DDC 00       3355          DEFB 0
0DDD 43414C43 3356  PNCALC: DEFM 'CALCULATOR'
0DE7 00       3357          DEFB 0
0DE8 53435249 3358  PNSCB:  DEFM 'SCRIBBLING'
0DF2 00       3359          DEFB 0
0DF3 53454C45 3360  GAMSTR: DEFM 'SELECT GAME'
0DFE 67       3361          DEFB 67H
0DFF 08       3362          DEFB 8
0E00 58       3363          DEFB 88
0E01 0D       3364          DEFB 1101B
0E02 28432920 3365          DEFM '(C) BALLY MFG 1977'
0E14 00       3366          DEFB 0
0E15          3367          END

TOTAL ASSEMBLER ERRORS =
                                  CROSS REFERENCE

LABEL       VALUE        REFERENCE

A0           00E1          -509
A1           0070          -521
A2           0037          -533
A3           001B          -545
A4           000D          -557
A5           0006          -563
ACTINT       000E          -226    227   3040   3040   3084
AKEYS        0214         -1123   1075   3041
ALKEYS       0214           -50
AS0          00D4          -510
AS1          006A          -522
AS2          0034          -534
AS3          001A          -546
B0           00C8          -511
B1           0064          -523
B2           0031          -535
B3           0018          -547
BCDAD        0321         -1315    942
BCDADD       0062          -278    279
BCDCHS       006A          -282    283   1324   1324   1333   1333
BCDCS        0364         -1391    946
BCDD0        0BEE         -2872   2994
BCDD1        0BFD         -2881   2980   2982
BCDD2        0C05         -2885   2996
BCDD3        0C0B         -2888   2990
BCDD4        0C11         -2891   2984
BCDDIV       0068          -281    282
BCDDV        0284         -1208    945
BCDISP       0BEB         -2870    920   2950
BCDML        02DE         -1268    944
BCDMUL       0066          -280    281
BCDNEG       006C          -283    284   1334   1334   1336   1336
BCDNG        0341         -1350    947
BCDNG1       034D         -1359   1388
BCDSB        031F         -1316    943
BCDSUB       0064          -279    280
BEGRAM       4FCE          -595    640   3067   3069
BITSPL       00A0           -44
BLANK        002A          -244    245
BMUSIC       0012          -230    231
BYTEPL       0028           -43   1506   2169   2248   2270   2289   2311   2348
                           2377   2611   3214   3217   3218
C1           00BD          -512
C2           005E          -524
C3           002E          -536
C4           0017          -548
C5           000B          -558
C6           0005          -564
C7           0002          -567
CALCL        01FA         -1099   3346
CALCST       1020          -652   1101
CBA          0009          -124    774   1080   1089   2129   2729   2932
CBB          0007          -122    842   1090   3245
```

```
CBC           0006         -121    843   1405   2032   2510   2557   2588   2949
CBD           0005         -120    775   2497   2499   2529   2784   3173
CBE           0004         -119    776   2530   2606   2783   3172
CBFLAG        0008         -123   1443   2028   2069
CBH           000B         -126    768   1270   2933
CBIXH         0003         -118    770    870
CBIXL         0002         -117    771    867
CBIYH         0001         -116
CBIYL         0000         -115
CBL           000A         -125    769   1271   2934
CCT1          03E6        -1516   1537
CCTLP         03DD        -1509   1538
CHDOWN        0001         -112
CHLEFT        0002         -111
CHRDIS        0032         -249    250   3121   3121   3123   3123   3151   3151
CHRIGH        0003         -110
CHTRIG        0004         -109
CHUP          0000         -113
CKSUM1        0033         -711
CKSUM2        0B0E        -2698
CLRNUM        0D99        -3147   3243   3296
CML           0DBE        -3178   1129
CMPIT         0C22        -2907   3008
CMPLOP        0C30        -2915   3028
CMSTRT        1328         -651   3348
CNT           4FDD         -612   1547   1675   1677
COL0L         0004         -169
COL0R         0000         -165
COL1L         0005         -170
COL1R         0001         -166
COL2L         0006         -171
COL2R         0002         -167
COL3L         0007         -172
COL3R         0003         -168
COLBX         000B         -173   1072   1084
COLLST        4FE8         -623   1082   1083
COLSET        0018         -235    236   3082
CONC1         0264        -1169   1159
CONC2         002B         -705   1171
CONCM         0008         -190    662
CONCPL        0256        -1158   1144   1153
CS1           00B2         -513
CS2           0059         -525
CS3           002C         -537
CS4           0015         -549
CS5           000A         -559
CT0           4FD5         -603   1660
CT1           4FD6         -604
CT2           4FD7         -605
CT3           4FD8         -606
CT4           4FD9         -607
CT5           4FDA         -608
CT6           4FDB         -609
CT7           4FDC         -610
CTIMER        0203          -47
CTLP          03D9        -1507   1549   1552
D1            00A8         -514
```

```
D2        0054          -526
D3        0029          -538
D4        0014          -550
DABS      0072          -286   287  1257  1257  1259  1259
DADD      006E          -284   285  1233  1233  1243  1243  1288  1288
                        1338  1338
DCLCT1    0849         -2494  2519
DCLCTB    083E         -2486  2527  2586
DECCTS    0010          -227   229
DELOAD    0074          -775  1580  2587
DISC1A    07F1         -2441  2470
DISC1B    07FE         -2448  2464
DISCH1    07ED         -2439  2461
DISCH2    080A         -2453  2481
DISCH3    080D         -2454  2479
DISCH4    0821         -2466  2500
DISCH5    0839         -2479  2471
DISNUM    0036          -251   253  3307  3307
DISPCH    07E1         -2433   918  2405  2955  2992
DISTIM    0052          -268   269
DIV1      029F         -1229  1248
DIV2      02A3         -1230  1236
DIV3      02B1         -1237  1233
DIV4      0315         -1307  1251
DOIT      0044          -261   262  3255  3255
DOITB     0046          -262   263
DS1       009F          -515
DS2       004F          -527
DS3       0027          -539
DS4       0013          -551
DS5       0009          -560
DS6       0004          -565
DSMG      0070          -285   286
DURAT     4FEA          -625  1691  1804  1911  1923
E1        0096          -516
E2        004A          -528
E3        0025          -540
E4        0012          -552
EMUSIC    0014          -231   233  3077
END       00C0          -380
ENDSCR    4FF4          -633  3017
ENTSTG    0DB7         -3176  3198
EPLOP     0410         -1543  1560
ETLP      0493         -1648  1663  1666
F1        008D          -517
F2        0046          -529
F3        0022          -541
F4        0011          -553
F5        0008          -561
FILL      001A          -236   237  3069  3069  3213  3213  3217  3217
                        3326  3326
FINDL3    0CF4         -3040  1076  2467  2502  2920
FIRSTC    2000           -41  3064  3066  3086
FNTSML    020D           -49
FNTSYS    0206           -48
FS1       0085          -518
FS2       0042          -530
```

```
FS3       0020       -542
FS4       0010       -554
FTBASE    0000        -94   2472
FTBYTE    0003        -97   2476   2494   2546
FTFSX     0001        -95   2531
FTFSY     0002        -96   2537
FTPTH     0006       -100   2482
FTPTL     0005        -99   2483
FTYSIZ    0004        -98   2477   2489
G0        00FD       -507
G1        007E       -519
G2        003E       -531
G3        001F       -543
G4        000F       -555
G5        0007       -562
G6        0003       -566
G7        0001       -568
G8        0000       -569
GAMSTB    4FF8       -635   1752   3014   3023
GAMSTR    0DF3      -3192   3085
GETNUM    004E       -266    267   3143   3143   3207   3207
GETPAR    004C       -265    266
GFSTRT    17DE       -650   1131
GMOVR     0C57      -2937   3038
GNACC     02C0      -1245   1208   1280
GNUMDO    0D4C      -3103   3255
GOUT      0502      -1732   1715   1747   1751   1754
GS0       00EE       -508
GS1       0077       -520
GS2       003B       -532
GS3       001D       -544
GS4       000E       -556
GSBEND    0007        -63   1755   3024
GSBSCR    0001        -62   3015
GSBTIM    0000        -61   1753
GTO1      04F4      -1724   1740
GTO2      04F9      -1728   1736
GTIMER    04E0      -1708   1724
GTMINS    4FEE       -629   2947
GTSECS    4FED       -628   2957
GUNLNK    0218      -1129   3091
HANDLE    0453      -1590   1606
HORAF     000F       -196
HORCB     0009       -174   1515
HUMANR    0040       -258    259
INCLOP    0C18      -2900   3010
INCSCR    0054       -269    271
INDEXB    005C       -275    276
INDEXN    0056       -272    273
INDEXW    005A       -274    275
INFBK     000D       -187   1045
INLIN     000F       -189   1043
INMOD     000E       -188   1519
INTPC     0000       -217    218   1232   1233   1242   1243   1257   1259
                     1288   1324   1333   1334   1336   1338   3035   3040
                     3041   3069   3076   3076   3076   3093   3121   3123
                     3126   3143   3151   3195   3201   3207   3208   3213
```

```
                    3217  3224  3252  3253  3255  3300  3305  3307
                    3326
INTPE     004E      -754   844
INTP@     0000     -2963 -2967
INTST     0008      -194
INXNIB    0B76     -2745   936
ITAB      0034      -713  1040  1044
JOYS      0471     -1610  1624
KCTASC    0040      -259   260  3300  3300
KCTATB    0AD5     -2640  2726
KEY0      0014      -207
KEY1      0015      -208
KEY2      0016      -209
KEY3      0017      -210  1582
KEYSEX    4FE3      -618  1570  1719
LRGCHR    08E4     -2610  1113
M81       053A     -1791  1826
M815      0540     -1794  1824
M82       0547     -1798  1816
M83       054B     -1801  1819  1821
MACTIN    018B     -1034   713   900
MAGIC     000C      -191  2127  2556  2616  2788
MATH      0056      -271   272
MBLAN1    07A3     -2328  2358
MBLAN2    07A4     -2329  2355
MBLANK    079E     -2324   914
MCALL     0006      -220   221
MCOLOR    01DB     -1083   905
MDISTI    0BCC     -2840   934
MDO1A     061C     -1935
MDOIT     060C     -1923   927
MDOIT0    060E     -1925  1961
MDOIT1    0616     -1931  1952
MDOIT2    0620     -1938  1958
MDOIT3    0621     -1939
MDOITB    060B     -1922   928
MENTRY    01AC     -1062   926
MENU      004A      -264   265  3093  3093
MENUCL    0013      -676  3082
MENUST    0218       -51
MFILL     0AEE     -2672   906
MFILL1    0AEF     -2673  2767
MFROG     0B06     -2693  2850
MGETN     0D31     -3094   932
MGETN1    0D36     -3097  3249
MGETN2    0D55     -3101  3259
MGETN3    0D61     -3110  3260
MGETN4    0D6F     -3121  3279
MGETN5    0D75     -3126  3289
MGETN6    0D7F     -3134  3258
MGETN7    0D86     -3138  3295
MGETN8    0D93     -3141  3292
MGETN9    0D96     -3142  3265  3301
MGETP     0CFB     -3058   931
MINCSC    0C15     -2898   935
MINDB     0BBD     -2824   939
MINDB1    0BC5     -2829  2921
```

```
MINDW      0BAC      -2807    938
MINT0      0084       -828    753
MINT1      0095       -837    835
MINT2      009A       -840    830
MINTPC     007B       -814    893
MJUMP      000A       -222    223
MKCTAS     0AC9      -2631    925
MMCALL     007D       -824    896   1968
MMENU      0C97      -2992    930
MMENU1     0CA2      -2996   3136
MMENU3     0CC8      -3012   3151
MMENU5     0CD9      -3019   3146
MMENU6     0CDF      -3020   3148
MMENU7     0CE2      -3023   3159
MMJUMP     0AC4      -2622    898
MMOVE      0B4B      -2701    940
MMRET      0B73      -2740    897
MMTD       0240      -1144    956
MMTD1      024E      -1152   1150
MMTD2      024F      -1153   1147
MNCLR      0D19      -3070   3114   3193
MNGH       0005      -2951
MNGL       0004      -2950
MNNH       0001      -2947   3131
MNNL       0000      -2946   3132
MNSAH      0003      -2949   3123
MNSAL      0002      -2948   3124
MO0        055B      -1809   1802
MO01       056B      -1818   1837
MO1        0574      -1823   1834
MO2        057D      -1828   1847
MO3        0587      -1834   1853
MO4        0594      -1841   1859
MO40       05A1      -1848   1865
MO41       05A5      -1850   1867
MO43       05B7      -1858   1876
MO44       05C0      -1865   1902
MO45       05C5      -1867   1857
MO5        05CC      -1870   1873
MO6        05DA      -1879   1895
MO61       05E6      -1884   1903
MOVE       005E       -276    277
MPAINT     06B2      -2099    907
MPAUSE     001B       -687    933
MPIZBK     01BA      -1068    929   1079
MPT1       06C5      -2112   2140
MPT2       06CF      -2117   2137
MPT3       06D5      -2122   2151
MPT4       06DE      -2128   2147
MQUIT      0C41      -2930    953
MQUIT1     0C4A      -2932   3045
MQUIT2     0C56      -2936   3043
MRANGE     037F      -1425    952
MRARGT     014B       -968    833
MRCALL     0632      -1952    895   1970
MRELA1     0AFB      -2689    922
MRELA2     0B00      -2691   2779
```

```
MRELAB    0AF6        -2686   921  2231
MREST     07AD        -2340   916
MREST1    07B5        -2347  2381
MRET      0008         -221   222
MRFLOP    0006         -102  1146  2235  2330
MRLOCK    4FF7         -634
MROR      0004         -104
MRROT     0002         -106
MRSHFT    0003         -107
MRXOR     0005         -103
MRXPND    0003         -105  2237  2276
MSAVE     03B9        -1469   915
MSAVE1    03C2        -1477  1509
MSCRL1    026B        -1179  1189
MSCROL    026A        -1178   917
MSENK2    043B        -1570  1586  1596
MSENKE    0446        -1579  1593
MSETB     036C        -1398   954
MSETUP    03CF        -1491   904
MSETW     0023         -697   955
MSHFTU    0DAA        -3165   941
MSK1      042C        -1560  1590
MSKTD     007E         -292
MSUCK     00A4         -857   899
MSUCK1    00A8         -863   839  2412
MSUCK2    00B6         -871   864
MSUCK3    00BF         -879   886
MSUCK5    00C6         -884   880
MULT1     02CD        -1251  1264
MULT2     02E1        -1269  1296
MULT3     02E8        -1275  1291
MULT4     02F0        -1279  1285
MULT5     0309        -1298  1314
MULT6     031B        -1308  1240  1322  1325
MULT7     0313        -1305  1316
MUZ999    05F4        -1893  1832
MUZAK     0012         -229   230
MUZCP1    0517        -1774  1768
MUZCPU    0514        -1773  1699
MUZPC     4FCE         -597  1797  1917
MUZSET    0508        -1741   902
MUZSP     4FD0         -598  1766  1798  1918
MUZSTP    05FC        -1898   903  1767  1909
MVBLA1    079A        -2315  2331
MVBLAN    077D        -2301   913
MVCT1A    066F        -2047  2068
MVECT     0633        -2004   924
MVECT1    0665        -2040  2065
MVECT2    0684        -2062  2080  2082
MVECT3    06A4        -2082  2085
MVECT6    06A6        -2084  2072
MVECTC    0656        -2033   923  2039
MVWRIT    06FE        -2174   908
MWRIT     0719        -2207   911
MWRITA    071C        -2211   912
MWRITP    0715        -2200   910
MWRITR    070B        -2184   909
```

```
MWRT       0725       -2217  2252
MWRTFL     074C       -2252  2236
MWX        0735       -2231  2238
MWX1       0736       -2232  2273
MWX2       0739       -2235  2265
MWXF       0766       -2272  2277
MWXF1      0767       -2273  2314
MWXF2      076A       -2276  2306
MXINTC     0279       -1194   894
MXSCR      021E         -52
NEGT       0074        -287   288  1232  1232  1242  1242
NOGAME     0235         -54
NOLINE     0060       -2945  3078  3218
NOPLAY     0228         -53
NORMEM     4000         -40  3213  3217
NUMBAS     000B        -669  3253
NUMPLY     4FF3        -632
NWHDWR     0001         -37  2836
NXTFR1     0858       -2507  2532
NXTFR2     0863       -2513  2538
NXTFR3     086A       -2517  2534
NXTFRM     084E       -2503  2466  2487
OA1        008F        -577
OA2        0047        -578
OA3        0023        -579
OA4        0011        -580
OA5        0008        -581
OBO        00FE        -571
OCO        00F1        -572
OD1        00D6        -573
OE1        00BF        -574
OF1        00B4        -575
OG1        00A0        -576
OPLOOP     051B       -1775  1841  1846  1851  1864
OPLP2      0592       -1840  1871  1881  1888  1893  1907
OPOTO      4FDF        -614
OPOT1      4FE0        -615
OPOT2      4FE1        -616
OPOT3      4FE2        -617
OSWO       4FE4        -619
OSW1       4FE5        -620
OSW2       4FE6        -621
OSW3       4FE7        -622
PAWS       0050        -267   268  3208  3208
PBLP       01C7       -1075
PFUG       0008        -649  1559
PHOT       040B       -1538  1558
PIZBRK     0048        -263   264
PNCALC     0DDD       -3188  1100
PNCM       0DD3       -3186  3347
PNGF       0DCA       -3184  1130
PNSCB      0DE8       -3190  3350
POTO       001C        -202  1093  1553  3281
POT1       001D        -203
POT2       001E        -204
POT3       001F        -205
PRIOR      4FF9        -636  1685  1756  1904  1906  1924
```

```
PSWCY       0000          -59
PSWPV       0002          -58
PSWSGN      0007          -56
PSWZRO      0006          -57   2028   2069
PUSH1       005D         -763    761
PUTNB1      0BA5        -2794   2889
PUTNB2      0BA8        -2796   2897
PUTNIB      0B90        -2778    937
PVOLAB      4FD2         -599   1828   1861
PVOLMC      4FD3         -600   1830
PWRUP       0C61        -2954    663
PWRUP1      0C95        -2974   3090
QFROG       0AD1        -2637   1474   2785
QUIT        0078         -289    290
R1          03A2        -1445   1473
R2          03A6        -1448   1470
R3          03A9        -1450   1467
RANGED      0076         -288    289   3252   3252
RANSHT      4FEF         -631   1450   1455   1456   1460
RCALL       0004         -219    220
RECTAN      001C         -237    238
RELAB1      003A         -254    255
RELABS      0038         -253    254
RELD        0068         -770   2413
RELTA       0B08        -2695   2592   2778
RELTA1      0B4E        -2705   2124   2781   2787
RELTA2      0B56        -2711
RELTA3      0B5F        -2717   2818
RENTER      007C         -815    827
REPEAT      0C3B        -2922   3021
RESTOR      002E         -246    247
RETN        027A        -1197    751
SAVE        002C         -245    246
SCBL        0DC4        -3181   1099
SCBST       0E19         -653   3351
SCHEDR      000C         -225    226
SCREEN      0000          -42
SCROLL      0030         -247    249
SCRSTR      0016         -233    234
SCT0        0001         -129
SCT1        0002         -130
SCT2        0003         -131
SCT3        0004         -132
SCT4        0005         -133
SCT5        0006         -134
SCT6        0007         -135
SCT7        0008         -136
SDABS       0356        -1374    950
SDADD       036E        -1408    948
SDADD1      036F        -1409   1439
SDSMG       0329        -1323    949
SDSMG1      0333        -1331   1360
SEMI4S      4FDE         -613
SENFLG      4FFA         -637   1062
SENTRY      0042         -260    261   3041   3041   3253   3253
SETB        007A         -290    291
SETEND      0C35        -2919   3029
```

```
SETOUT     0016        -234   235  3078
SETW       007C        -291   292
SF0        0009        -137
SF1        000A        -138
SF2        000B        -139
SF3        000C        -140
SF4        000D        -141
SF5        000E        -142
SF6        000F        -143
SF7        0010        -144
SH1        03B1       -1455  1482
SHFTU1     0DB1       -3171  3341
SHIFTR     03AC       -1451  1451  1458
SHIFTU     0060        -277   278  3305  3305
SIXY       04CC       -1689  1694  1702  1706  1709
SJ0        0015        -153
SJ1        0017        -155
SJ2        0019        -157
SJ3        001B        -159
SKYD       0013        -146  1091  1602  3044  3258
SKYU       0012        -147  1592
SMLCHR     0ABF       -2620  1120
SMLFNT     020D       -1115  2945
SNDBX      0018        -185  1806  1839  1925
SNEGT      034C       -1358   951
SNUL       0000        -128
SP0        001C        -148  3260
SP1        001D        -149
SP2        001E        -150
SP3        001F        -151
SSEC       0011        -145  1576
ST0        0014        -152  3042  3259
ST1        0016        -154
ST2        0018        -156
ST3        001A        -158
STAKD      04BE       -1679  1696
STHLDE     0BB8       -2816  1155
STIMER     0200         -46
STOREN     0058        -273   274
STRD1      07CE       -2381  2402
STRD2      07D4       -2384  2404
STRDIS     0034        -250   251  3035  3035  3126  3126  3195  3195
                       3201  3201  3224  3224
STRIPE     06E2       -2134  2139  2152
STRNEW     07C4       -2375   919  2407  2414
STRP1      06EB       -2139  2174
SUCK       000C        -223   225
SW0        0010        -198  1614
SW1        0011        -199
SW2        0012        -200
SW3        0013        -201
SWHIT      0461       -1599  1618
SWLOP      0456       -1591  1620
SYSDPT     00CB        -893   756
SYSFNT     0206       -1108  2462
SYSRAM     4FCE        -640
TIMEX      047B       -1625  1106
```

```
TIMEY     047E       -1635   901  1048
TIMEZ     04A0       -1660  1046  1105
TIMLP     0485       -1638  1674
TIMOUT    4FEC        -627  1065  1096  3074
TKEYS     0421       -1555  1573
TMR60     4FEB        -626
TONEA     0011        -178
TONEB     0012        -179
TONEC     0013        -180
TONMO     0010        -177
TPLOP     03FF       -1530  1568
TRCHK     03EC       -1522  1088
TSEX      0413       -1546  1546
TTEST     01E5       -1088  1067  1077
UMARGT    4FFB        -638   836
UPISTR    0000        -216   217
USERTB    4FFD        -639   762
VBBLNK    0006         -88  2201  2325  2327
VBCCHK    0004         -85  2071  2090  2093  2111
VBCH      0003         -84  2060  2088  2109
VBCL      0002         -83  2061  2089  2108
VBCLAT    0003         -92  2090  2111
VBCLMT    0000         -90  2071
VBCREV    0001         -91  2093
VBDCH     0001         -82  2058  2104
VBDCL     0000         -81  2059  2103
VBDXH     0004         -69
VBDXL     0003         -68  2037  2040
VBDYH     0009         -74
VBDYL     0008         -73  2040
VBLANK    0028        -243   244
VBMR      0000         -65  2198  2330
VBOAH     000E         -79  2328
VBOAL     000D         -78  2329
VBSACT    0007         -87  2029
VBSTAT    0001         -66  2029  2201  2325  2327
VBTIMB    0002         -67  2030  2031
VBXCHK    0007         -72
VBXH      0006         -71  2200
VBXL      0005         -70
VBYCHK    000C         -77
VBYH      000B         -76  2199
VBYL      000A         -75
VECT      003E        -256   258
VECTC     003C        -255   256
VERAF     000E        -195
VERBL     000A        -175
VIBRA     0014        -181
VOICES    4FD4        -601  1765  1805  1850
VOLAB     0016        -182  1071  1711  1829  1914
VOLC      0015        -183  1070  1712  1831  1915
VOLN      0017        -184
VWRITR    001E        -238   239
WASTE     0FFF        -586   587  2160  2161  3072
WASTER    0FFF        -587
WRFL1     0751       -2255  2293
WRFL2     0754       -2258  2285
```

```
WRIT      0024        -241   242
WRITA     0026        -242   243
WRITP     0022        -240   241
WRITR     0020        -239   240
WRTL1     088D       -2538  2584
WRTL2     0898       -2546  2576
WRTL3     08AC       -2562  2559
WRTL4     08BF       -2570  2617
WRTL5     08C4       -2575  2605
WRTL6     08D4       -2586  2608
WRTLIN    086C       -2522  2492
XINTC     0002        -218   219  3085
XNIB      0B7B       -2756  2849  2979
XNIB1     0B8C       -2769  2868
XPAND     0019        -192  2554  2589
XPNDON    0001         -36
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

               641
               642            LIST S,X,M,T
               643  ; **********
               644  ; * SKETCH *
               645  ; **********
               646  ;
               647  ;THE OFFICIAL NAME OF THIS
               648  ;PROGRAM IS SCRIBBLING
               649  ;
               650  ; SKETCH EQUATES
               651  ; SKETCH PACKET DISPLACEMENTS:
>001E          652  SCPSIZ  EQU  30              ; SIZE OF SKETCH PACKET
>0000          653  SCSAVA  EQU  0               ; SAVE AREA START
>001A          654  SCXC    EQU  26              ; X COORDINATE
>001B          655  SCYC    EQU  27              ; Y COORDINATE
>001C          656  SCSADL  EQU  28              ; SAVE ADDRESS LO AND HI
>001D          657  SCSADH  EQU  29
               658  ; OTHER GOODIES
>0004          659  MOVTMR  EQU  4               ; MOVE RATE
>0014          660  KSCTRV  EQU  20              ; COLOR STEPPING TIME
               661          ORG  0E19H           ; ** START
 0E19          662  BEGIN:  SYSSUK GETPAR
 0E19 FF       662 +        RST  56
 0E1A 4D       662 +        DEFB GETPAR+1
               662 +        IF   GETPAR.EQ.INTPC
               662 +        ENDIF
 0E1B 2802     663          DEFW NOPLAY
 0E1D 01       664          DEFB 1
 0E1E F34F     665          DEFW NUMPLY
 0E20          666  SCCLR:
 0E20 31E84E   667          LD   SP,SCRSTK
 0E23          668          SYSTEM INTPC
 0E23 FF       668 +        RST  56
 0E24 00       668 +        DEFB INTPC
               668 +        IF   INTPC.EQ.INTPC
>0001          668 +INTP@   DEFL 1
               668 +        ENDIF
 0E25          669          DO   FILL            ; CLEAR SCREEN
 0E25 1B       669 +        DEFB FILL+1
 0E26 0040     670          DEFW NORMEM
 0E28 600E     671          DEFW 92*BYTEPL
 0E2A 00       672          DEFB 0
 0E2B          673          DO   FILL
 0E2B 1B       673 +        DEFB FILL+1
 0E2C F04E     674          DEFW P1SCP
 0E2E 7800     675          DEFW SCPSIZ*4
 0E30 00       676          DEFB 0
 0E31          677          DO   SETOUT
 0E31 17       677 +        DEFB SETOUT+1
 0E32 B8       678          DEFB 184
 0E33 28       679          DEFB 40
 0E34 08       680          DEFB 8
 0E35          681          DO   MOVE
 0E35 5F       681 +        DEFB MOVE+1
 0E36 E84E     682          DEFW COLORS
 0E38 0800     683          DEFW 8
 0E3A 0C10     684          DEFW INICOL
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

0E3C           685          DO   COLSET
0E3C 19        685 +        DEFB COLSET+1
0E3D E84E      686          DEFW COLORS
0E3F           687          DO   SETW
0E3F 7D        687 +        DEFB SETW+1
0E40 46        688          DEFB 70
0E41 24        689          DEFB 36
0E42 0A4F      690          DEFW P1SCP+SCXC
0E44           691          DO   SETW
0E44 7D        691 +        DEFB SETW+1
0E45 53        692          DEFB 83
0E46 24        693          DEFB 36
0E47 284F      694          DEFW P2SCP+SCXC
0E49           695          DO   SETW
0E49 7D        695 +        DEFB SETW+1
0E4A 46        696          DEFB 70
0E4B 30        697          DEFB 48
0E4C 464F      698          DEFW P3SCP+SCXC
0E4E           699          DO   SETW
0E4E 7D        699 +        DEFB SETW+1
0E4F 53        700          DEFB 83
0E50 30        701          DEFB 48
0E51 644F      702          DEFW P4SCP+SCXC
0E53           703          DO   SETB
0E53 7B        703 +        DEFB SETB+1
0E54 04        704          DEFB MOVTMR
0E55 D54F      705          DEFW CTO
0E57           706          DONT XINTC
0E57 02        706 +        DEFB XINTC
0E58 21580E    707  MAINLP: LD   HL,MAINLP
0E5B E5        708          PUSH HL
0E5C           709          SYSSUK SENTRY
0E5C FF        709 +        RST  56
0E5D 43        709 +        DEFB SENTRY+1
               709 +        IF   SENTRY.EQ.INTPC
               709 +        ENDIF
0E5E 650E      710          DEFW KEYMES
0E60           711          SYSSUK DOIT
0E60 FF        711 +        RST  56
0E61 45        711 +        DEFB DOIT+1
               711 +        IF   DOIT.EQ.INTPC
               711 +        ENDIF
0E62 A10E      712          DEFW SCDOTB
0E64 C9        713          RET
0E65 2F        714  KEYMES: DEFB 2FH
0E66 0F        715          DEFB 0FH
0E67 0F        716          DEFB 0FH
0E68 0F        717          DEFB 0FH
               718  ; KEYBOARD HANDLER
0E69 05        719  KEYBO:  DEC  B
0E6A 0E03      720          LD   C,3
0E6C 78        721          LD   A,B
0E6D FE14      722          CP   20           ; CLEAR ENTRY DOWN?
0E6F 28AF      723          JR   Z,SCCLR-$    ; JUMP TO CLEAR IF SO
0E71 0F        724          RRCA
0E72 0F        725          RRCA
0E73 A1        726          AND  C
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

0E74           727           SYSSUK INDEXB
0E74 FF        727 +         RST   56
0E75 5D        727 +         DEFB INDEXB+1
               727 +         IF    INDEXB.EQ.INTPC
               727 +         ENDIF
0E76 290F      728           DEFW CDELTB
0E78 EB        729           EX    DE,HL
0E79 78        730           LD    A,B
0E7A A1        731           AND   C
0E7B 67        732           LD    H,A
0E7C 79        733           LD    A,C
0E7D 94        734           SUB   H
0E7E           735           SYSSUK INDEXB       ; POINT AT COLOR
0E7E FF        735 +         RST   56
0E7F 5D        735 +         DEFB INDEXB+1
               735 +         IF    INDEXB.EQ.INTPC
               735 +         ENDIF
0E80 E84E      736           DEFW COLORS
0E82 1A        737           LD    A,(DE)
0E83 86        738           ADD   A,(HL)        ; ADD DELTA FACTOR
0E84 CB58      739           BIT   3,B           ; WAS KEY FOR INTENSITY?
0E86 2804      740           JR    Z,KEYB1-$
0E88 AE        741           XOR   (HL)
0E89 E607      742           AND   7
0E8B AE        743           XOR   (HL)
0E8C 77        744  KEYB1:   LD    (HL),A
0E8D 23        745           INC   HL            ; CHANGE COLOR ON OTHER SIDE
0E8E 23        746           INC   HL
0E8F 23        747           INC   HL
0E90 23        748           INC   HL
0E91 77        749           LD    (HL),A
0E92           750           SYSSUK COLSET
0E92 FF        750 +         RST   56
0E93 19        750 +         DEFB COLSET+1
               750 +         IF    COLSET.EQ.INTPC
               750 +         ENDIF
0E94 E84E      751           DEFW COLORS
0E96 3E14      752           LD    A,KSCTRV      ; SET KEYSEX CLEAR TIMER
0E98 32D64F    753           LD    (CT1),A
0E9B C9        754           RET
               755  ; ROUTINE TO CLEAR KEYSEX
0E9C AF        756  KLRKSX:  XOR   A
0E9D 32E34F    757           LD    (KEYSEX),A
0EA0 C9        758           RET
0EA1           759  SCDOTB:  JMP   SCTO,DOWRTS
0EA1 01        759 +         DEFB SCTO
0EA2 D30F      759 +         DEFW DOWRTS
               759 +         IF    0
               759 +         ENDIF
0EA4           760           JMP   SCT1,KLRKSX
0EA4 02        760 +         DEFB SCT1
0EA5 9C0E      760 +         DEFW KLRKSX
               760 +         IF    0
               760 +         ENDIF
0EA7           761           JMP   SKYD,KEYBO
0EA7 13        761 +         DEFB SKYD
0EA8 690E      761 +         DEFW KEYBO
```

```
               761 +        IF   0
               761 +        ENDIF
               762  ; ITERATE THROUGH ACTIVE PLAYERS SUBROUTINE
0EAA DD21F04E  763  ITER4:  LD   IX,P1SCP
0EAE 3AF34F    764          LD   A,(NUMPLY)
0EB1 47        765          LD   B,A
0EB2 4F        766          LD   C,A
0EB3 C5        767  ITER41: PUSH BC
0EB4 E5        768          PUSH HL
0EB5 11BA0E    769          LD   DE,ITRET
0EB8 D5        770          PUSH DE
0EB9 E9        771          JP   (HL)
0EBA 111E00    772  ITRET:  LD   DE,SCPSIZ
0EBD DD19      773          ADD  IX,DE
0EBF E1        774          POP  HL
0EC0 C1        775          POP  BC
0EC1 10F0      776          DJNZ ITER41-$
0EC3 C9        777          RET
               778  ; UPDATE COORDINATES ROUTINE
0EC4 79        779  SCRUPD: LD   A,C
0EC5 90        780          SUB  B
0EC6           781          SYSSUK INDEXB
0EC6 FF        781 +        RST  56
0EC7 5D        781 +        DEFB INDEXB+1
               781 +        IF   INDEXB.EQ.INTPC
               781 +        ENDIF
0EC8 E44F      782          DEFW OSWO
0ECA E60F      783          AND  0FH
0ECC CD0110    784          CALL GETDLT       ; GET DELTAS
0ECF DD7E1A    785          LD   A,(IX+SCXC)  ; UPDATE X
0ED2 82        786          ADD  A,D
0ED3 FE98      787          CP   152          ; OUT OF BOUNDS?
0ED5 3003      788          JR   NC,SCRUP1-$
0ED7 DD771A    789          LD   (IX+SCXC),A
0EDA DD7E1B    790  SCRUP1: LD   A,(IX+SCYC)  ; SAME FOR Y
0EDD 84        791          ADD  A,H
0EDE FE55      792          CP   85
0EE0 D0        793          RET  NC
0EE1 DD771B    794          LD   (IX+SCYC),A
0EE4 C9        795          RET
               796  ; RESTORE
0EE5 DDE5      797  SCREST: PUSH IX
0EE7 D1        798          POP  DE
0EE8 1A        799          LD   A,(DE)
0EE9 A7        800          AND  A
0EEA C8        801          RET  Z
0EEB DD661D    802          LD   H,(IX+SCSADH)
0EEE DD6E1C    803          LD   L,(IX+SCSADL)
0EF1           804          SYSTEM RESTOR
0EF1 FF        804 +        RST  56
0EF2 2E        804 +        DEFB RESTOR
               804 +        IF   RESTOR.EQ.INTPC
               804 +        ENDIF
0EF3 AF        805          XOR  A
0EF4 12        806          LD   (DE),A
0EF5 C9        807          RET
               808  ; WRITE ROUTINE
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

0EF6 79         809  SCRWRT:  LD   A,C
0EF7 90         810           SUB  B
0EF8            811           SYSSUK INDEXB
0EF8 FF         811 +         RST  56
0EF9 5D         811 +         DEFB INDEXB+1
                811 +         IF   INDEXB.EQ.INTPC
                811 +         ENDIF
0EFA E44F       812           DEFW OSWO
0EFC E610       813           AND  10H
0EFE C8         814           RET  Z
0EFF 2B         815  SCRWR1:  DEC  HL              ; BACKUP TO POT
0F00 2B         816           DEC  HL
0F01 2B         817           DEC  HL
0F02 2B         818           DEC  HL
0F03 2B         819           DEC  HL
0F04 7E         820           LD   A,(HL)
0F05 07         821           RLCA
0F06 07         822           RLCA
0F07 4F         823           LD   C,A
0F08 E603       824           AND  3
0F0A            825           SYSSUK INDEXB       ; SET SIZES
0F0A FF         825 +         RST  56
0F0B 5D         825 +         DEFB INDEXB+1
                825 +         IF   INDEXB.EQ.INTPC
                825 +         ENDIF
0F0C 260F       826           DEFW SIZTBL
0F0E DD561B     827           LD   D,(IX+SCYC)
0F11 DD5E1A     828           LD   E,(IX+SCXC)
0F14 47         829           LD   B,A
0F15 79         830           LD   A,C
0F16 07         831           RLCA
0F17 07         832           RLCA
0F18 E603       833           AND  3
0F1A            834  SCRWR2:  SYSSUK INDEXB
0F1A FF         834 +         RST  56
0F1B 5D         834 +         DEFB INDEXB+1
                834 +         IF   INDEXB.EQ.INTPC
                834 +         ENDIF
0F1C 220F       835           DEFW COLMSK
0F1E 48         836           LD   C,B
0F1F            837           SYSTEM RECTAN
0F1F FF         837 +         RST  56
0F20 1C         837 +         DEFB RECTAN
                837 +         IF   RECTAN.EQ.INTPC
                837 +         ENDIF
0F21 C9         838           RET
0F22 00         839  COLMSK:  DEFB 0
0F23 55         840           DEFB 01010101B
0F24 AA         841           DEFB 10101010B
0F25 FF         842           DEFB 11111111B
0F26 01         843  SIZTBL:  DEFB 1
0F27 02         844           DEFB 2
0F28 04         845           DEFB 4
0F29 08         846  CDELTB:  DEFB 8
0F2A F8         847           DEFB -8
0F2B 01         848           DEFB 1
0F2C FF         849           DEFB -1
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

              850   ; SAVE ROUTINE
0F2D 78       851   SCRSAV: LD   A,B
0F2E          852           SYSSUK INDEXB
0F2E FF       852 +         RST  56
0F2F 5D       852 +         DEFB INDEXB+1
              852 +         IF   INDEXB.EQ.INTPC
              852 +         ENDIF
0F30 E34F     853           DEFW OSWO-1
0F32 E610     854           AND  10H
0F34 C0       855           RET  NZ
0F35 E5       856           PUSH HL
0F36 DD561B   857           LD   D,(IX+SCYC)
0F39 DD5E1A   858           LD   E,(IX+SCXC)
0F3C          859           SYSTEM RELAB1
0F3C FF       859 +         RST  56
0F3D 3A       859 +         DEFB RELAB1
              859 +         IF   RELAB1.EQ.INTPC
              859 +         ENDIF
0F3E DD721D   860           LD   (IX+SCSADH),D
0F41 DD731C   861           LD   (IX+SCSADL),E
0F44 EB       862           EX   DE,HL
0F45 DDE5     863           PUSH IX
0F47 D1       864           POP  DE
0F48 010308   865           LD   BC,0803H     ; SAVE WORST CASE
0F4B          866           SYSTEM SAVE
0F4B FF       866 +         RST  56
0F4C 2C       866 +         DEFB SAVE
              866 +         IF   SAVE.EQ.INTPC
              866 +         ENDIF
0F4D E1       867           POP  HL
0F4E 18AF     868           JR   SCRWR1-$
              869   ; ZERO PLAYER GAME WRITE HANDLER
0F50 21F04E   870   ZEROPL: LD   HL,ZPSTMR    ; LOAD PTR TO SIZE TIMER
0F53 11F34E   871           LD   DE,ZPSIZ     ; AND SIZE TRACKER
0F56 35       872           DEC  (HL)         ; DECREMENT SIZE TIMER
0F57 F2690F   873           JP   P,ZPA        ; JUMP IF NO COUNTDOWN
0F5A          874           SYSSUK RANGED     ; GET NEW SIZE
0F5A FF       874 +         RST  56
0F5B 77       874 +         DEFB RANGED+1
              874 +         IF   RANGED.EQ.INTPC
              874 +         ENDIF
0F5C 30       875           DEFB 48
0F5D FE08     876           CP   8            ; 8-47?
0F5F 3802     877           JR   C,ZPO-$      ; NO - ZPO
0F61 E603     878           AND  3            ; YES - HAVE MORE 1-4S
0F63 3C       879   ZPO:    INC  A
0F64 12       880           LD   (DE),A       ; SET NEW SIZE
0F65          881           SYSSUK RANGED     ; GET NEW SIZE TIMER
0F65 FF       881 +         RST  56
0F66 77       881 +         DEFB RANGED+1
              881 +         IF   RANGED.EQ.INTPC
              881 +         ENDIF
0F67 78       882           DEFB 120
0F68 77       883           LD   (HL),A
0F69 23       884   ZPA:    INC  HL           ; ADVANCE TO COLOR STUFF
0F6A 13       885           INC  DE
0F6B 35       886           DEC  (HL)         ; AND DEC COLOR TIMER
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND       COMMENT

0F6C F2770F    887          JP   P,ZPB
0F6F           888          SYSSUK RANGED      ; GET NEW COLOR
0F6F FF        888 +        RST  56
0F70 77        888 +        DEFB RANGED+1
               888 +        IF   RANGED.EQ.INTPC
               888 +        ENDIF
0F71 04        889          DEFB 4
0F72 12        890          LD   (DE),A
0F73           891          SYSSUK RANGED      ; GET NEW COLOR TIMER
0F73 FF        891 +        RST  56
0F74 77        891 +        DEFB RANGED+1
               891 +        IF   RANGED.EQ.INTPC
               891 +        ENDIF
0F75 78        892          DEFB 120
0F76 77        893          LD   (HL),A
0F77 23        894  ZPB:    INC  HL            ; TO DIRECTION STUFF
0F78 13        895          INC  DE
0F79 35        896          DEC  (HL)          ; DECREMENT DIRECTION TIMER
0F7A F2930F    897          JP   P,ZPD
0F7D 11F54E    898  ZPC:    LD   DE,DIRVAL     ; DE = DIRECTION TRACKER
0F80           899          SYSSUK RANGED      ; DRAW NEW DIRECTION
0F80 FF        899 +        RST  56
0F81 77        899 +        DEFB RANGED+1
               899 +        IF   RANGED.EQ.INTPC
               899 +        ENDIF
0F82 0A        900          DEFB 10
0F83 3C        901          INC  A
0F84 FE03      902          CP   3             ; REJECT ILLEGAL VALUES
0F86 28F5      903          JR   Z,ZPC-$
0F88 FE07      904          CP   7
0F8A 28F1      905          JR   Z,ZPC-$
0F8C 12        906          LD   (DE),A
0F8D           907          SYSSUK RANGED
0F8D FF        907 +        RST  56
0F8E 77        907 +        DEFB RANGED+1
               907 +        IF   RANGED.EQ.INTPC
               907 +        ENDIF
0F8F 28        908          DEFB 40
0F90 32F24E    909          LD   (DIRTMR),A
0F93 1A        910  ZPD:    LD   A,(DE)        ; GET DIRECTION VALUE
0F94 CD0110    911          CALL GETDLT        ; GET DELTAS
0F97 010A4F    912          LD   BC,P1SCP+SCXC ; POINT AT COORDINATES
0F9A 0A        913          LD   A,(BC)
0F9B 82        914          ADD  A,D
0F9C FE50      915          CP   80
0F9E 30DD      916          JR   NC,ZPC-$      ; GET NEW DIRECTION IF AT LMT
0FA0 02        917          LD   (BC),A
0FA1 5F        918          LD   E,A           ; SAVE X COORDINATE
0FA2 03        919          INC  BC
0FA3 0A        920          LD   A,(BC)
0FA4 84        921          ADD  A,H
0FA5 FE2E      922          CP   46
0FA7 30D4      923          JR   NC,ZPC-$
0FA9 02        924          LD   (BC),A
0FAA 57        925          LD   D,A           ; SET Y COORDINATE
0FAB 21F34E    926          LD   HL,ZPSIZ      ; POINT AT SIZES AGAIN
0FAE 46        927          LD   B,(HL)
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND          COMMENT

0FAF 23         928           INC   HL
0FB0 7E         929           LD    A,(HL)          ; GET COLOR TOO
0FB1 CD1A0F     930           CALL  SCRWR2          ; DO FIRST WRITE
0FB4 67         931           LD    H,A             ; SAVE COLOR
0FB5 D5         932           PUSH  DE              ; AND X,Y
0FB6 3E5C       933           LD    A,92            ; REFLECT Y
0FB8 90         934           SUB   B
0FB9 92         935           SUB   D
0FBA 57         936           LD    D,A
0FBB 7C         937           LD    A,H
0FBC            938           SYSTEM RECTAN
0FBC FF         938 +         RST   56
0FBD 1C         938 +         DEFB  RECTAN
                938 +         IF    RECTAN.EQ.INTPC
                938 +         ENDIF
0FBE 3EA0       939           LD    A,160           ; REFLECT X
0FC0 91         940           SUB   C
0FC1 93         941           SUB   E
0FC2 5F         942           LD    E,A
0FC3 7C         943           LD    A,H
0FC4            944           SYSTEM RECTAN
0FC4 FF         944 +         RST   56
0FC5 1C         944 +         DEFB  RECTAN
                944 +         IF    RECTAN.EQ.INTPC
                944 +         ENDIF
0FC6 E1         945           POP   HL              ; RESTORE X,Y
0FC7 54         946           LD    D,H             ; RESTORE Y
0FC8            947           SYSTEM RECTAN
0FC8 FF         947 +         RST   56
0FC9 1C         947 +         DEFB  RECTAN
                947 +         IF    RECTAN.EQ.INTPC
                947 +         ENDIF
0FCA 3EFF       948           LD    A,0FFH          ; RESET TIMEOUT
0FCC 32EC4F     949           LD    (TIMOUT),A
0FCF 3E01       950  ZERO1:   LD    A,1             ; RESET COUNTER-TIMER
0FD1 182A       951           JR    ZERO2-$
0FD3 3AF34F     952  DOWRTS:  LD    A,(NUMPLY)
0FD6 3D         953           DEC   A
0FD7 FE04       954           CP    4
0FD9 D2500F     955           JP    NC,ZEROPL
0FDC 21C40E     956           LD    HL,SCRUPD
0FDF CDAA0E     957           CALL  ITER4
0FE2 21E50E     958           LD    HL,SCREST
0FE5 CDAA0E     959           CALL  ITER4
0FE8 21F60E     960           LD    HL,SCRWRT
0FEB CDAA0E     961           CALL  ITER4
                962  ; NOW GOING BACKWARDS SAVE AND WRITE EVERYBODY WITH TRIGG
0FEE 41         963           LD    B,C
0FEF 11E2FF     964  SCRB3:   LD    DE,-SCPSIZ
0FF2 DD19       965           ADD   IX,DE
0FF4 C5         966           PUSH  BC
0FF5 CD2D0F     967           CALL  SCRSAV
0FF8 C1         968           POP   BC
0FF9 10F4       969           DJNZ  SCRB3-$
0FFB 3E04       970           LD    A,MOVTMR
0FFD 32D54F     971  ZERO2:   LD    (CTO),A
1000 C9         972           RET                   ; DONE
```

```
 ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

                973   ; SUBROUTINE TO SCARE UP DELTAS
 1001 C5        974   GETDLT: PUSH BC
 1002 47        975           LD   B,A
 1003           976           SYSSUK MSKTD
 1003 FF        976 +         RST  56
 1004 7F        976 +         DEFB MSKTD+1
                976 +         IF   MSKTD.EQ.INTPC
                976 +         ENDIF
 1005 0001      977           DEFW 100H
 1007 00        978           DEFB 0
 1008 0001      979           DEFW 100H
 100A C1        980           POP  BC
 100B C9        981           RET
                982   ; INITIAL COLORS:
 100C 08        983   INICOL: DEFB 08H
 100D 5B        984           DEFB 5BH
 100E A5        985           DEFB 0A5H
 100F 07        986           DEFB 007H
 1010 08        987           DEFB 08H
 1011 5B        988           DEFB 5BH
 1012 A5        989           DEFB 0A5H
 1013 07        990           DEFB 07H
                991           ORG  4000H+3720
                992   ; SKETCH RAM:
 4E88           993           DEFS 96
 4EE8           994   SCRSTK:
 4EE8           995   COLORS: DEFS 8
>4EF0           996   ZPSTMR  EQU  $
>4EF2           997   DIRTMR  EQU  ZPSTMR+2
>4EF3           998   ZPSIZ   EQU  DIRTMR+1
>4EF5           999   DIRVAL  EQU  ZPSIZ+2
 4EF0          1000   P1SCP:  DEFS SCPSIZ
 4F0E          1001   P2SCP:  DEFS SCPSIZ
 4F2C          1002   P3SCP:  DEFS SCPSIZ
 4F4A          1003   P4SCP:  DEFS SCPSIZ
 4F68          1004           END

TOTAL ASSEMBLER ERRORS =
                               CROSS REFERENCE

LABEL      VALUE        REFERENCE

A0          00E1          -508
A1          0070          -520
A2          0037          -532
A3          001B          -544
A4          000D          -556
A5          0006          -562
ACTINT      000E          -225
ALKEYS      0214           -49
AS0         00D4          -509
AS1         006A          -521
AS2         0034          -533
AS3         001A          -545
B0          00C8          -510
B1          0064          -522
B2          0031          -534
B3          0018          -546
BCDADD      0062          -277
BCDCHS      006A          -281
BCDDIV      0068          -280
BCDMUL      0066          -279
BCDNEG      006C          -282
BCDSUB      0064          -278
BEGIN       0E19          -662
BEGRAM      4FCE          -594
BITSPL      00A0           -43
BLANK       002A          -243
BMUSIC      0012          -229
BYTEPL      0028           -42    671
C1          00BD          -511
C2          005E          -523
C3          002E          -535
C4          0017          -547
C5          000B          -557
C6          0005          -563
C7          0002          -566
CBA         0009          -123
CBB         0007          -121
CBC         0006          -120
CBD         0005          -119
CBE         0004          -118
CBFLAG      0008          -122
CBH         000B          -125
CBIXH       0003          -117
CBIXL       0002          -116
CBIYH       0001          -115
CBIYL       0000          -114
CBL         000A          -124
CDELTB      0F29          -804    728
CHDOWN      0001          -111
CHLEFT      0002          -110
CHRDIS      0032          -248
CHRIGH      0003          -109
CHTRIG      0004          -108
```

```
CHUP      0000      -112
CNT       4FDD      -611
COL0L     0004      -168
COL0R     0000      -164
COL1L     0005      -169
COL1R     0001      -165
COL2L     0006      -170
COL2R     0002      -166
COL3L     0007      -171
COL3R     0003      -167
COLBX     000B      -172
COLLST    4FE8      -622
COLMSK    0F22      -797   835
COLORS    4EE8      -927   682    686    736    751
COLSET    0018      -234   686    751    751
CONCM     0008      -189
CS1       00B2      -512
CS2       0059      -524
CS3       002C      -536
CS4       0015      -548
CS5       000A      -558
CT0       4FD5      -602   705    971
CT1       4FD6      -603   753
CT2       4FD7      -604
CT3       4FD8      -605
CT4       4FD9      -606
CT5       4FDA      -607
CT6       4FDB      -608
CT7       4FDC      -609
CTIMER    0203       -46
D1        00A8      -513
D2        0054      -525
D3        0029      -537
D4        0014      -549
DABS      0072      -285
DADD      006E      -283
DECCTS    0010      -226
DIRTMR    4EF2      -929   909    998
DIRVAL    4EF5      -931   898
DISNUM    0036      -250
DISTIM    0052      -267
DOIT      0044      -260   712    712
DOITB     0046      -261
DOWRTS    0FD3      -886   760
DS1       009F      -514
DS2       004F      -526
DS3       0027      -538
DS4       0013      -550
DS5       0009      -559
DS6       0004      -564
DSMG      0070      -284
DURAT     4FEA      -624
E1        0096      -515
E2        004A      -527
E3        0025      -539
E4        0012      -551
EMUSIC    0014      -230
```

```
END        00C0        -379
ENDSCR     4FF4        -632
F1         008D        -516
F2         0046        -528
F3         0022        -540
F4         0011        -552
F5         0008        -560
FILL       001A        -235    670    674
FIRSTC     2000         -40
FNTSML     020D         -48
FNTSYS     0206         -47
FS1        0085        -517
FS2        0042        -529
FS3        0020        -541
FS4        0010        -553
FTBASE     0000         -93
FTBYTE     0003         -96
FTFSX      0001         -94
FTFSY      0002         -95
FTPTH      0006         -99
FTPTL      0005         -98
FTYSIZ     0004         -97
G0         00FD        -506
G1         007E        -518
G2         003E        -530
G3         001F        -542
G4         000F        -554
G5         0007        -561
G6         0003        -565
G7         0001        -567
G8         0000        -568
GAMSTB     4FF8        -634
GETDLT     1001        -908    784    911
GETNUM     004E        -265
GETPAR     004C        -264    663    663
GS0        00EE        -507
GS1        0077        -519
GS2        003B        -531
GS3        001D        -543
GS4        000E        -555
GSBEND     0007         -62
GSBSCR     0001         -61
GSBTIM     0000         -60
GTMINS     4FEE        -628
GTSECS     4FED        -627
HORAF      000F        -195
HORCB      0009        -173
HUMANR     0040        -257
INCSCR     0054        -268
INDEXB     005C        -274    728    728    736    736    782    782    812
                        812    826    826    835    835    853    853
INDEXN     0056        -271
INDEXW     005A        -273
INFBK      000D        -186
INICOL     100C        -915    684
INLIN      000F        -188
INMOD      000E        -187
```

```
INTPC     0000       -216   663   669   669   669   710   712   728
                      736   751   782   805   812   826   835   838
                      853   860   867   875   882   889   892   900
                      908   939   945   948   977
INTP@     0001       -666
INTST     0008       -193
ITER4     0EAA       -733   957   959   961
ITER41    0EB3       -737   776
ITRET     0EBA       -742   769
KCTASC    0040       -258
KEY0      0014       -206
KEY1      0015       -207
KEY2      0016       -208
KEY3      0017       -209
KEYB0     0E69       -701   762
KEYB1     0E8C       -722   740
KEYMES    0E65       -696   710
KEYSEX    4FE3       -617   757
KLRKSX    0E9C       -732   761
KSCTRV    0014       -660   752
MAGIC     000C       -190
MAINLP    0E58       -693   707
MATH      0056       -270
MCALL     0006       -219
MENU      004A       -263
MENUST    0218        -50
MJUMP     000A       -221
MOVE      005E       -275   682
MOVTMR    0004       -659   704   970
MRET      0008       -220
MRFLOP    0006       -101
MRLOCK    4FF7       -633
MROR      0004       -103
MRROT     0002       -105
MRSHFT    0003       -106
MRXOR     0005       -102
MRXPND    0003       -104
MSKTD     007E       -291   977   977
MUZAK     0012       -228
MUZPC     4FCE       -596
MUZSP     4FD0       -597
MXSCR     021E        -51
NEGT      0074       -286
NOGAME    0235        -53
NOPLAY    0228        -52   663
NORMEM    4000        -39   670
NUMPLY    4FF3       -631   665   764   952
NWHDWR    0001        -36
OA1       008F       -576
OA2       0047       -577
OA3       0023       -578
OA4       0011       -579
OA5       0008       -580
OB0       00FE       -570
OC0       00F1       -571
OD1       00D6       -572
OE1       00BF       -573
```

```
OF1       00B4        -574
OG1       00A0        -575
OPOT0     4FDF        -613
OPOT1     4FE0        -614
OPOT2     4FE1        -615
OPOT3     4FE2        -616
OSW0      4FE4        -618  782   812   853
OSW1      4FE5        -619
OSW2      4FE6        -620
OSW3      4FE7        -621
P1SCP     4EF0        -932  674   690   763   912
P2SCP     4F0E        -933  694
P3SCP     4F2C        -934  698
P4SCP     4F4A        -935  702
PAWS      0050        -266
PIZBRK    0048        -262
POT0      001C        -201
POT1      001D        -202
POT2      001E        -203
POT3      001F        -204
PRIOR     4FF9        -635
PSWCY     0000         -58
PSWPV     0002         -57
PSWSGN    0007         -55
PSWZRO    0006         -56
PVOLAB    4FD2        -598
PVOLMC    4FD3        -599
QUIT      0078        -288
RANGED    0076        -287  875   875   882   882   889   889   892
                       892  900   900   908   908
RANSHT    4FEF        -630
RCALL     0004        -218
RECTAN    001C        -236  838   838   939   939   945   945   948
                       948
RELAB1    003A        -253  860   860
RELABS    0038        -252
RESTOR    002E        -245  805   805
SAVE      002C        -244  867   867
SCCLR     0E20        -664  723
SCDOTB    0EA1        -735  712
SCHEDR    000C        -224
SCPSIZ    001E        -652  675   772   964  1000  1001  1002  1003
SCRB3     0FEF        -898  969
SCREEN    0000         -41
SCREST    0EE5        -765  958
SCROLL    0030        -246
SCRSAV    0F2D        -809  967
SCRSTK    4EE8        -926  667
SCRSTR    0016        -232
SCRUP1    0EDA        -758  788
SCRUPD    0EC4        -749  956
SCRWR1    0EFF        -779  868
SCRWR2    0F1A        -796  930
SCRWRT    0EF6        -775  960
SCSADH    001D        -657  802   860
SCSADL    001C        -656  803   861
SCSAVA    0000        -653
```

```
SCT0      0001        -128   760
SCT1      0002        -129   761
SCT2      0003        -130
SCT3      0004        -131
SCT4      0005        -132
SCT5      0006        -133
SCT6      0007        -134
SCT7      0008        -135
SCXC      001A        -654   690   694   698   702   785   789   828
                       858   912
SCYC      001B        -655   790   794   827   857
SEMI4S    4FDE        -612
SENFLG    4FFA        -636
SENTRY    0042        -259   710   710
SETB      007A        -289   704
SETOUT    0016        -233   678
SETW      007C        -290   688   692   696   700
SF0       0009        -136
SF1       000A        -137
SF2       000B        -138
SF3       000C        -139
SF4       000D        -140
SF5       000E        -141
SF6       000F        -142
SF7       0010        -143
SHIFTU    0060        -276
SIZTBL    0F26        -801   826
SJ0       0015        -152
SJ1       0017        -154
SJ2       0019        -156
SJ3       001B        -158
SKYD      0013        -145   762
SKYU      0012        -146
SNDBX     0018        -184
SNUL      0000        -127
SP0       001C        -147
SP1       001D        -148
SP2       001E        -149
SP3       001F        -150
SSEC      0011        -144
ST0       0014        -151
ST1       0016        -153
ST2       0018        -155
ST3       001A        -157
STIMER    0200         -45
STOREN    0058        -272
STRDIS    0034        -249
SUCK      000C        -222
SW0       0010        -197
SW1       0011        -198
SW2       0012        -199
SW3       0013        -200
SYSRAM    4FCE        -639
TIMOUT    4FEC        -626   949
TMR60     4FEB        -625
TONEA     0011        -177
TONEB     0012        -178
```

```
TONEC       0013         -179
TONMO       0010         -176
UMARGT      4FFB         -637
UPISTR      0000         -215
USERTB      4FFD         -638
VBBLNK      0006          -87
VBCCHK      0004          -84
VBCH        0003          -83
VBCL        0002          -82
VBCLAT      0003          -91
VBCLMT      0000          -89
VBCREV      0001          -90
VBDCH       0001          -81
VBDCL       0000          -80
VBDXH       0004          -68
VBDXL       0003          -67
VBDYH       0009          -73
VBDYL       0008          -72
VBLANK      0028         -242
VBMR        0000          -64
VBOAH       000E          -78
VBOAL       000D          -77
VBSACT      0007          -86
VBSTAT      0001          -65
VBTIMB      0002          -66
VBXCHK      0007          -71
VBXH        0006          -70
VBXL        0005          -69
VBYCHK      000C          -76
VBYH        000B          -75
VBYL        000A          -74
VECT        003E         -255
VECTC       003C         -254
VERAF       000E         -194
VERBL       000A         -174
VIBRA       0014         -180
VOICES      4FD4         -600
VOLAB       0016         -181
VOLC        0015         -182
VOLN        0017         -183
VWRITR      001E         -237
WASTE       0FFF         -585
WASTER      0FFF         -586
WRIT        0024         -240
WRITA       0026         -241
WRITP       0022         -239
WRITR       0020         -238
XINTC       0002         -217   707
XPAND       0019         -191
XPNDON      0001          -35
ZERO1       0FCF         -884
ZERO2       0FFD         -905   951
ZEROPL      0F50         -822   955
ZPO         0F63         -829   877
ZPA         0F69         -832   873
ZPB         0F77         -838   887
ZPC         0F7D         -842   903    905    916    923
```

```
ZPD       0F93        -850   897
ZPSIZ     4EF3        -930   871   926   999
ZPSTMR    4EF0        -928   870   997
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                641
                642           LIST S,X,M,T
                643           NLIST I
                644           ;*****************************
                645           ;* H V G   C H E C K M A T E *
                646           ;*****************************
                647           ;
                648           ;
                649           ; M A C R O S
                650           ;
                651  DEF4X:   MACR #A4X,#B4X,#C4X,#D4X
                652           DEFB #A4X
                653           DEFB #B4X
                654           DEFB #C4X
                655           DEFB #D4X
                656           ENDM
                657  WRECK    MACR
                658           DEFW 9.SHL.8+32
                659           DEFB 0000B
                660           ENDM
                661           ;
                662           ;
                663           ; E Q U A T E S
                664           ;
>0000           665  OLDWAY   EQU  1-NWHDWR      ;1=DO OLD WAY 0=DO NEW WAY
>0001           666  NEWWAY   EQU  1-OLDWAY      ;OPPOSITE OF OLDWAY
                667           ;VARIOUS EQU'S
>000C           668  RLMOVE   EQU  1100B         ;RIGHT AND LEFT MOVES
>0003           669  UDMOVE   EQU  0011B         ;UP AND DOWN MOVES
>0002           670  NGBIT    EQU  2             ;# OF GAMES BIT
>0003           671  NPBIT    EQU  3             ;# PLAYERS BIT
>0003           672  ANIMAX   EQU  3             ;MAX # TICKS PER ANIMATION FRAM
>009C           673  XMAX     EQU  (BYTEPL-1)*4  ;MAX X COORD
>0015           674  YLINES   EQU  21            ;# VERT BLOCKS
>000B           675  LOWY     EQU  11            ;LOWEST Y COORD
>005B           676  YMAX     EQU  ((YLINES-1)*4)+LOWY ;MAX Y COORD
>0000           677  LOWX     EQU  0             ;LOWEST X COORD
>0008           678  AMOVE    EQU  8H            ;AN ARBITRARY MOVE
>0009           679  MUSVOL   EQU  09H           ;MUSIC VOLUME
>0024           680  TDOPT    EQU  100100B       ;TIME DISPLAY OPTIONS
>0044           681  CDOPT    EQU  01000100B     ;COUNT DOWN OPT
>0010           682  WRITOR   EQU  010000B       ;WRIT WITH MAGIC OR
                683           ;PLAYER PACKET OFFSETS
>0000           684  LASTSW   EQU  0             ;LAST SWITCH SETTING
>0001           685  LASTMV   EQU  1             ;LAST ACTUAL MOVE
>0002           686  CURSW    EQU  2             ;CURRENT SWITCH SETTING
>0003           687  AROT     EQU  3             ;ARROW ROTATION AMOUNT
>0004           688  ARRX     EQU  4             ;ARROW X COORD
>0005           689  ARRY     EQU  5             ;ARROW Y COORD
>0006           690  PSTAT    EQU  6             ;PLAYER STATUS
                691           ;PLAYER STATUS MASKS
>0080           692  ACTIVE   EQU  80H
>0040           693  HUMAN    EQU  40H
>0007           694  ACTBIT   EQU  7             ; 1=ACTIVE 0=DEAD
>0006           695  HUMBIT   EQU  6             ; 1=HUMAN 0=COMPUTER
                696           ;SCREEN TABS
>0028           697  XTAB1    EQU  ((BYTEPL/4)*4)
```

```
>0050           698  XTAB2    EQU  XTAB1*2
>0078           699  XTAB3    EQU  XTAB1*3
>0014           700  YTAB     EQU  (((YLINES-1)/4)*4)
>001F           701  YTAB1    EQU  YTAB+LOWY
>0033           702  YTAB2    EQU  (2*YTAB)+LOWY
>0047           703  YTAB3    EQU  (3*YTAB)+LOWY
                704           ;OFFSETS FOR EACH PLAYERS ROM DATA
>0000           705  NOTE0    EQU  0              ;EACH DIRECTIONS NOTES
>0001           706  NOTE1    EQU  1
>0002           707  NOTE2    EQU  2
>0003           708  NOTE3    EQU  3
>0004           709  PPATL    EQU  4              ;PLAYER PAT ADDR LOW
>0005           710  PPATH    EQU  5              ;PLAYER PAT ADDR HIGH
>0006           711  PCDOP    EQU  6              ;PLAYER CHAR DISP OPT
>0007           712  PSPOSX   EQU  7              ;X COORD OF PLAYER SCORE
>0008           713  PSPOSY   EQU  8              ;Y COORD OF PLAYER SCORE
>0009           714  PSDOP    EQU  9              ;PLAYER SCORE DISP OPT
                715           ;MORE EQU'S
>00F6           716  FORCEM   EQU  0F6H           ;VAL TO FORCE RANDOM MOVE
>0004           717  WIDTH    EQU  4H             ;# PIXELS WIDE OF PLAYER PAT
>0004           718  HEIGHT   EQU  4H             ;# PIXELS HIGH OF PLAYER PAT
>0D20           719  ALLBYT   EQU  (YLINES*4)*BYTEPL ;ALL BYTES ON A SCREEN
>41B8           720  STARTS   EQU  (LOWY*BYTEPL)+NORMEM ;LOWEST ADDR OF PLAY FI
>0001           721  PATXSZ   EQU  1              ;#BYTES WIDE OF PLAYER PATTERN
>0004           722  PATYSZ   EQU  4              ;#BYTES HIGH OF PLAYER PATTERN
>0104           723  PATDIM   EQU  PATXSZ.SHL.8.OR.PATYSZ ;PATTERNS DIMENSIONS
>000F           724  JUSJOY   EQU  0FH            ;ONLY JOY STICK BITS
>0008           725  CBLEN    EQU  8              ;COLOR BLOCK LENGTH
>0008           726  SBLEN    EQU  8              ;SOUND BLOCK LENGTH
>0000           727  WPONOF   EQU  0
>0001           728  WPOPT    EQU  1
>0002           729  WPPAL    EQU  2
>0003           730  WPPAH    EQU  3
>0005           731  WPXSIZ   EQU  5
>0004           732  WPYSIZ   EQU  4
                733           ;
                734           ;
                735           ;
                736           ORG  NORMEM+0F96H ;SHOULD BE EQUAL TO RSTART
                737           ;UNCLEARED RAM
 4F96           738  UNCRAM:
 4F96           739  CURSCR:  DEFS 12             ;ALL CURRENT SCORES
                740           ;CLEARED RAM
 4FA2           741  CNOPL:   DEFS 1              ;CURRENT # PLAYERS
 4FA3           742  PLIX:    DEFS 1              ;WHO IS CURRENT PLAYER
 4FA4           743  CNOHUM:  DEFS 1              ;CURRENT # HUMANS
 4FA5           744  TARRX:   DEFS 1              ;TEMP ARROW X COORD
 4FA6           745  TARRY:   DEFS 1              ;TEMP ARROW Y COORD
 4FA7           746  RMASK:   DEFS 1              ;ROTATE MASK
                747  PPACKS:                      ;START OF PLAYER PACKETS
 4FA8           748  PLAY0:   DEFS PSTAT+1
 4FAF           749  PLAY1:   DEFS PSTAT+1
 4FB6           750  PLAY2:   DEFS PSTAT+1
 4FBD           751  PLAY3:   DEFS PSTAT+1
 4FC4           752  ENDRAM:
>4FA1           753  RSTART   EQU  BEGRAM-(ENDRAM-UNCRAM)+1 ;  SHOULD BE RAM STA
                754           ORG  1328H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1328           755   ONETIM:
               756            ;ONE TIME ONLY HOUSEKEEPING
1328 31964F    757            LD   SP,UNCRAM
132B           758            SYSSUK GETPAR
132B FF        758 +          RST  56
132C 4D        758 +          DEFB GETPAR+1
               758 +          IF   GETPAR.EQ.INTPC
               758 +          ENDIF
132D 3502      759            DEFW NOGAME
132F 82        760            DEFB 82H
1330 DC4F      761            DEFW CT7
1332           762            SYSSUK GETPAR
1332 FF        762 +          RST  56
1333 4D        762 +          DEFB GETPAR+1
               762 +          IF   GETPAR.EQ.INTPC
               762 +          ENDIF
1334 2802      763            DEFW NOPLAY
1336 01        764            DEFB 1
1337 F34F      765            DEFW NUMPLY
1339           766            SYSSUK FILL
1339 FF        766 +          RST  56
133A 1B        766 +          DEFB FILL+1
               766 +          IF   FILL.EQ.INTPC
               766 +          ENDIF
133B 964F      767            DEFW CURSCR
133D 0C00      768            DEFW 12
133F 00        769            DEFB 0
1340           770   FIREIT:
               771            ;RE-ENTRY POINT FROM END OF GAME
1340 F3        772            DI
1341 31964F    773            LD   SP,UNCRAM
1344           774            SYSTEM INTPC
1344 FF        774 +          RST  56
1345 00        774 +          DEFB INTPC
               774 +          IF   INTPC.EQ.INTPC
>0001          774 +INTP@     DEFL 1
               774 +          ENDIF
               775            ;OUTPUT COLOR BLOCK
1346           776            DO   COLSET
1346 19        776 +          DEFB COLSET+1
1347 AA17      777            DEFW CBLOCK
1349           778            DO   EMUSIC
1349 15        778 +          DEFB EMUSIC+1
               779            ;CLEAR JOY STICKS
134A           780            DO   FILL
134A 1B        780 +          DEFB FILL+1
134B E44F      781            DEFW OSWO
134D 0400      782            DEFW 4
134F 00        783            DEFB 0
               784            ;CLEAR ALL RAM DATA
1350           785            DO   FILL
1350 1B        785 +          DEFB FILL+1
1351 A24F      786            DEFW CNOPL
1353 2200      787            DEFW .RES.(PLAY3+PSTAT)-CNOPL+1
1355 00        788            DEFB 0
1356           789            DO   SETOUT
1356 17        789 +          DEFB SETOUT+1
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1357 BE         790           DEFB . RES. ((YLINES*4)+LOWY)*2 ;VER BLK
1358 40         791           DEFB 40H+0          ;HOR COL BND
1359 08         792           DEFB 08H            ;INTER MODE
                793           ;CLEAR SCORE BLOCKS
135A            794           DO   RECTAN
135A 1D         794 +         DEFB RECTAN+1
135B 0000       795           DEFW 0
135D A00B       796           DEFW 11. SHL. 8+160
135F 55         797           DEFB 01010101B
1360            798           DO   RECTAN
1360 1D         798 +         DEFB RECTAN+1
1361 8000       799           DEFW 0. SHL. 8+128
1363            800           WRECK
1363 2009       800 +         DEFW 9. SHL. 8+32
1365 00         800 +         DEFB 0000B
1366            801           DO   RECTAN
1366 1D         801 +         DEFB RECTAN+1
1367 5800       802           DEFW 0. SHL. 8+88
1369            803           WRECK
1369 2009       803 +         DEFW 9. SHL. 8+32
136B 00         803 +         DEFB 0000B
136C            804           DO   RECTAN
136C 1D         804 +         DEFB RECTAN+1
136D 2800       805           DEFW 0. SHL. 8+40
136F            806           WRECK
136F 2009       806 +         DEFW 9. SHL. 8+32
1371 00         806 +         DEFB 0000B
1372            807           DO   RECTAN
1372 1D         807 +         DEFB RECTAN+1
1373 0000       808           DEFW 0. SHL. 8+0
1375            809           WRECK
1375 2009       809 +         DEFW 9. SHL. 8+32
1377 00         809 +         DEFB 0000B
1378            810           DO   ACTINT
1378 0F         810 +         DEFB ACTINT+1
1379            811           EXIT
1379 02         811 +         DEFB XINTC
>0000           811 +INTPC    DEFL 0
                812           ;INITIALIZE STARTING ADDRESS OF ARROWS
137A 212833     813           LD   HL,. RES. (YTAB2. SHL. 8)+XTAB1
137D 22AC4F     814           LD   (PLAY0+ARRX),HL
1380 217833     815           LD   HL,. RES. (YTAB2. SHL. 8)+XTAB3
1383 22B34F     816           LD   (PLAY1+ARRX),HL
1386 21501F     817           LD   HL,. RES. (YTAB1. SHL. 8)+XTAB2
1389 22BA4F     818           LD   (PLAY2+ARRX),HL
138C 215047     819           LD   HL,. RES. (YTAB3. SHL. 8)+XTAB2
138F 22C14F     820           LD   (PLAY3+ARRX),HL
                821           ;CLEAR FIELD
1392 CDB414     822           CALL CLEARF
                823           ;DISPLAY # GAMES
1395 DD210D02   824           LD   IX,FNTSML
1399            825           SYSSUK DISNUM
1399 FF         825 +         RST  56
139A 37         825 +         DEFB DISNUM+1
                825 +         IF   DISNUM. EQ. INTPC
                825 +         ENDIF
139B 4C         826           DEFB 76
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

139C 02         827           DEFB 2
139D 24         828           DEFB TDOPT
139E 42         829           DEFB 42H
139F DC4F       830           DEFW CT7
13A1            831  DONTD:
                832           ;GET # HUMANS
13A1 3AF34F     833           LD   A,(NUMPLY)
13A4 FE05       834           CP   5
13A6 3802       835           JR   C,GOTNPL-$
13A8 3E04       836           LD   A,4
13AA            837  GOTNPL:
13AA 32A44F     838           LD   (CNOHUM),A
                839           ;GET # PLAYERS:
                840           ;  IF HUMANS=1 OR 0 OR > 4 THEN PLAYERS=4 ELSE PLA
13AD FE02       841           CP   2
13AF 3804       842           JR   C,FPLAY-$
13B1 FE05       843           CP   5
13B3 3802       844           JR   C,ALLHUM-$
13B5 3E04       845  FPLAY:   LD   A,4
13B7 32A24F     846  ALLHUM:  LD   (CNOPL),A
                847           ;INITIALIZE THE PLAYER PACKETS
                848           ;B=CURR # HUMANS
                849           ;C=CURR # PLAYERS
                850           ;D=THIS PLAYER #
13BA 3AA44F     851  INTIPP:  LD   A,(CNOHUM)
13BD 47         852           LD   B,A
13BE 3AA24F     853           LD   A,(CNOPL)
13C1 4F         854           LD   C,A
13C2 1600       855           LD   D,0
13C4 7A         856  GTPLIX:  LD   A,D
13C5 CD5C16     857           CALL LDPLIX
13C8 C5         858           PUSH BC
13C9 D5         859           PUSH DE
13CA 7A         860           LD   A,D
13CB C631       861           ADD  A,31H          ;SET UP ASCII LITERAL
13CD DD5E04     862           LD   E,(IX+ARRX)
13D0 DD5605     863           LD   D,(IX+ARRY)
13D3 1D         864           DEC  E
13D4 1D         865           DEC  E
13D5 FD4E06     866           LD   C,(IY+PCDOP)
13D8            867           SYSTEM CHRDIS       ;DISPLAY PLAYER# ON FIELD
13D8 FF         867 +         RST  56
13D9 32         867 +         DEFB CHRDIS
                867 +         IF   CHRDIS.EQ.INTPC
                867 +         ENDIF
13DA FD5E07     868           LD   E,(IY+PSPOSX)
13DD FD5608     869           LD   D,(IY+PSPOSY)
13E0 D5         870           PUSH DE
13E1            871           SYSTEM CHRDIS       ;DISPLAY# ON SCORE BLOCK
13E1 FF         871 +         RST  56
13E2 32         871 +         DEFB CHRDIS
                871 +         IF   CHRDIS.EQ.INTPC
                871 +         ENDIF
13E3 D1         872           POP  DE
13E4 7B         873           LD   A,E
13E5 C606       874           ADD  A,6
13E7 5F         875           LD   E,A
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

13E8 14         876            INC  D
13E9 14         877            INC  D
13EA 010104     878            LD   BC,PATYSZ.SHL.8+PATXSZ
13ED FD6605     879            LD   H,(IY+PPATH)
13F0 FD6E04     880            LD   L,(IY+PPATL)
13F3 3E10       881            LD   A,00010000B
13F5            882            SYSTEM WRIT          ;WRIT PLAYER PAT IN SCORE BLOCK
13F5 FF         882 +          RST  56
13F6 24         882 +          DEFB WRIT
                882 +          IF   WRIT.EQ.INTPC
                882 +          ENDIF
13F7 D1         883            POP  DE
13F8 D5         884            PUSH DE
13F9 DDE5       885            PUSH IX
13FB 7A         886            LD   A,D
13FC 0600       887            LD   B,0
13FE 4A         888            LD   C,D
13FF 21964F     889            LD   HL,CURSCR
1402 09         890            ADD  HL,BC
1403 09         891            ADD  HL,BC
1404 09         892            ADD  HL,BC
1405 CDE315     893            CALL DISPSC          ;DISPLAY SCORES
1408 DDE1       894            POP  IX
140A D1         895            POP  DE
140B C1         896            POP  BC
140C AF         897            XOR  A
140D B0         898            OR   B
140E 2809       899            JR   Z,NOTHUM-$
1410 3EC0       900            LD   A,ACTIVE+HUMAN
1412 DD7706     901            LD   (IX+PSTAT),A
1415 05         902            DEC  B
1416 1806       903            JR   CKNOPL-$
1418 00         904  CKSUM3:   DEFB 0               ;
1419 3E80       905  NOTHUM:   LD   A,ACTIVE
141B DD7706     906            LD   (IX+PSTAT),A
141E 14         907  CKNOPL:   INC  D
141F 0D         908            DEC  C
1420 AF         909            XOR  A
1421 B1         910            OR   C
1422 20A0       911            JR   NZ,GTPLIX-$
1424 3E03       912            LD   A,3
1426            913  CDOWNL:
1426 F5         914            PUSH AF
1427            915            SYSSUK PAWS
1427 FF         915 +          RST  56
1428 51         915 +          DEFB PAWS+1
                915 +          IF   PAWS.EQ.INTPC
                915 +          ENDIF
1429 05         916            DEFB 5
142A 32A34F     917            LD   (PLIX),A
142D CD9114     918            CALL UPMUZK          ;MAKE SOUND FOR COUNT DOWN
1430 F1         919            POP  AF
1431 F5         920            PUSH AF
1432 C630       921            ADD  A,30H
1434            922            XYRELL DE,(XTAB2-4),.RES.(YTAB2-4)
1434 00000000   922 +          LD   DE,.RES.(.RES.(YTAB2-4)).SHL.8+((XTAB2-4))
1438 0E44       923            LD   C,CDOPT
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

143A           924            SYSTEM CHRDIS      ;DISPLAY COUNT DOWN #
143A FF        924 +          RST  56
143B 32        924 +          DEFB CHRDIS
               924 +          IF   CHRDIS. EQ. INTPC
               924 +          ENDIF
143C           925            SYSSUK PAWS
143C FF        925 +          RST  56
143D 51        925 +          DEFB PAWS+1
               925 +          IF   PAWS. EQ. INTPC
               925 +          ENDIF
143E 28        926            DEFB 40
143F           927            SYSTEM EMUSIC
143F FF        927 +          RST  56
1440 14        927 +          DEFB EMUSIC
               927 +          IF   EMUSIC. EQ. INTPC
               927 +          ENDIF
1441 F1        928            POP  AF
1442 3D        929            DEC  A
1443 20E1      930            JR   NZ,CDOWNL-$
1445 CDB414    931            CALL CLEARF
               932            ;INIT TICK COUNT
1448 CD4A16    933            CALL TICKIT
144B AF        934            XOR  A
144C 32DD4F    935            LD   (CNT),A
144F           936   LOOPY:
144F           937            SYSSUK SENTRY
144F FF        937 +          RST  56
1450 43        937 +          DEFB SENTRY+1
               937 +          IF   SENTRY. EQ. INTPC
               937 +          ENDIF
1451 1402      938            DEFW ALKEYS
1453           939            SYSSUK DOIT
1453 FF        939 +          RST  56
1454 45        939 +          DEFB DOIT+1
               939 +          IF   DOIT. EQ. INTPC
               939 +          ENDIF
1455 5914      940            DEFW THETBL
1457 18F6      941            JR   LOOPY-$
1459           942   THETBL:  RC   SCTO,ACTION
1459 41        942 +          DEFB SCTO+40H
145A 6C14      942 +          DEFW ACTION
               942 +          IF   0
               942 +          ENDIF
145C           943            RC   SJ0,MOVJOY
145C 55        943 +          DEFB SJ0+40H
145D 8414      943 +          DEFW MOVJOY
               943 +          IF   0
               943 +          ENDIF
145F           944            RC   SJ1,MOVJOY
145F 57        944 +          DEFB SJ1+40H
1460 8414      944 +          DEFW MOVJOY
               944 +          IF   0
               944 +          ENDIF
1462           945            RC   SJ2,MOVJOY
1462 59        945 +          DEFB SJ2+40H
1463 8414      945 +          DEFW MOVJOY
               945 +          IF   0
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                945 +         ENDIF
1465            946           RC   SJ3,MOVJOY
1465 5B         946 +         DEFB SJ3+40H
1466 8414       946 +         DEFW MOVJOY
                946 +         IF   0
                946 +         ENDIF
1468            947           RC   SKYD,CALPIZ,+END
1468 53         947 +         DEFB SKYD+40H
1469 8B14       947 +         DEFW CALPIZ
                947 +         IF   0+END
146B C0         947 +         DEFB 0+END
                947 +         ENDIF
146C            948   ACTION:
146C CD4A16     949           CALL TICKIT
                950           ;INCREMENT THE CURRENT PLAYER INDEX BY 1 UNTIL
                951           ;AN ACTIVE PLAYER IS FOUND THEN UPDATE HIM
146F 3AA34F     952   INCIX:  LD   A,(PLIX)
1472 3C         953           INC  A
1473 E603       954           AND  03H
1475 32A34F     955           LD   (PLIX),A      ;CURR PLAYER IX<-CURR PL IX+1 M
1478 CD5C16     956           CALL LDPLIX
147B DDCB067E   957           BIT  ACTBIT,(IX+PSTAT) ;TEST FOR ACTIVE PLAYER
147F 28EE       958           JR   Z,INCIX-$
1481 C3BC14     959           JP   MOVEIT        ;THE MAJOR EVENT
1484            960   MOVJOY:
1484 D615       961           SUB  SJ0           ;TAKE OFF WHATEVER
1486 CB3F       962           SRL  A             ;DIV BY 2
1488 C31E16     963           JP   STALL
148B            964   CALPIZ:
148B CD4A16     965           CALL TICKIT
148E            966           SYSTEM PIZBRK
148E FF         966 +         RST  56
148F 48         966 +         DEFB PIZBRK
                966 +         IF   PIZBRK.EQ.INTPC
                966 +         ENDIF
1490 C9         967           RET
1491 3AA34F     968   UPMUZK: LD   A,(PLIX)
1494 CD5C16     969           CALL LDPLIY
1497 DD7E03     970           LD   A,(IX+AROT)
149A 0603       971           LD   B,3
149C            972   TSTBIT:
149C 0F         973           RRCA
149D 3802       974           JR   C,GOTBIT-$
149F 10FB       975           DJNZ TSTBIT-$
14A1            976   GOTBIT:
14A1 48         977           LD   C,B
14A2 0600       978           LD   B,0
14A4 FD09       979           ADD  IY,BC
14A6 FD7E00     980           LD   A,(IY+0)
14A9 D313       981           OUT  (TONEC),A
14AB 3E09       982           LD   A,MUSVOL
14AD D315       983           OUT  (VOLC),A
14AF 3E11       984           LD   A,0A4
14B1 D310       985           OUT  (TONMO),A
14B3 C9         986           RET
14B4            987   CLEARF:
                988           ;CLEAR FIELD
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

14B4            989           SYSSUK FILL
14B4 FF         989 +         RST   56
14B5 1B         989 +         DEFB  FILL+1
                989 +         IF    FILL.EQ.INTPC
                989 +         ENDIF
14B6 B841       990           DEFW  STARTS
14B8 200D       991           DEFW  ALLBYT
14BA 00         992           DEFB  0
14BB C9         993           RET
14BC            994  MOVEIT:
                995           ;THIS ROUTINE UPDATES A PLAYER'S POSITION
                996           ;INPUT PARAS ARE:  IX=POINTER TO PLAYERS PACKET
                997           ;DURING ROUTINE B=CURRENT SWITCH C=LAST SWITCH
14BC DD4E00     998           LD    C,(IX+LASTSW)
14BF DD4602     999           LD    B,(IX+CURSW)
14C2 DDCB0676  1000           BIT   HUMBIT,(IX+PSTAT)
14C6 2003      1001           JR    NZ,NOCUR-$      ;IF NOT HUMAN
14C8 AF        1002  ZSW:     XOR   A               ; CLEAR A
14C9 47        1003           LD    B,A             ;CLEAR CURRENT SWITCH
14CA 4F        1004           LD    C,A             ;CLEAR LAST SW ENDIF
14CB 78        1005  NOCUR:   LD    A,B             ;IF CURR SW = 0
14CC B7        1006           OR    A
14CD 2001      1007           JR    NZ,RANTST-$
14CF 41        1008           LD    B,C             ;THEN CURR SW<-LAST SW ENDIF
14D0 DD7000    1009  RANTST:  LD    (IX+LASTSW),B ;SAVE LAST SW
14D3 78        1010           LD    A,B             ;IF CURR SW=0
14D4 B7        1011           OR    A
14D5 2005      1012           JR    NZ,GOTSW-$
14D7 0E00      1013           LD    C,0             ;LAST SW<-0
14D9 CD7F16    1014           CALL  RANMOV          ;GET RANDOM MOVE ENDIF
14DC           1015  GOTSW:
14DC DD7E01    1016           LD    A,(IX+LASTMV) ;GET LAST MOVE
14DF CDAC16    1017           CALL  MOVTST
14E2 2813      1018           JR    Z,GOTMOV-$
               1019                                ;ANY MOVE AND CURR SW
14E4 CDAA16    1020           CALL  MOVANY
14E7 280E      1021           JR    Z,GOTMOV-$
14E9 41        1022           LD    B,C             ;TRY LAST SWITCH
               1023                                ;ANY MOVE
14EA CDAA16    1024           CALL  MOVANY
14ED 2808      1025           JR    Z,GOTMOV-$
14EF DD4601    1026           LD    B,(IX+LASTMV) ;TRY LAST MOVE
               1027                                ;ANY MOVE
14F2 CDAA16    1028           CALL  MOVANY
14F5 203C      1029           JR    NZ,CRASH-$      ;
14F7           1030  GOTMOV:
               1031           ;A LEGIT MOVE HAS BEEN FOUND SO UPDATE THE GUY
14F7 DD7701    1032           LD    (IX+LASTMV),A ;SAVE ACTUAL MOVE FOR LATER
14FA DD7703    1033           LD    (IX+AROT),A   ;ARROW ROTATION AMOUNT<-THE MOV
14FD DD5605    1034           LD    D,(IX+ARRY)
1500 DD5E04    1035           LD    E,(IX+ARRX)
1503 CD2515    1036           CALL  ERASE
1506 FD6605    1037           LD    H,(IY+PPATH)
1509 FD6E04    1038           LD    L,(IY+PPATL)
150C 010104    1039           LD    BC,PATYSZ.SHL.8+PATXSZ
150F 3E10      1040           LD    A,WRITOR
1511           1041           SYSTEM WRIT           ;WRITE PLAYER PATTERN OVER ARRO
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

1511 FF       1041 +         RST  56
1512 24       1041 +         DEFB WRIT
              1041 +         IF   WRIT.EQ.INTPC
              1041 +         ENDIF
1513 3AA54F   1042           LD   A,(TARRX)
1516 DD7704   1043           LD   (IX+ARRX),A   ;SAVE NEW ARROW X
1519 3AA64F   1044           LD   A,(TARRY)
151C DD7705   1045           LD   (IX+ARRY),A   ;SAVE NEW ARROW Y
151F CD0016   1046           CALL ANIARR        ;ANIMATE THE ARROW
1522 C39114   1047           JP   UPMUZK
1525          1048  ERASE:
1525 D5       1049           PUSH DE
1526          1050           SYSSUK RELAB1
1526 FF       1050 +         RST  56
1527 3B       1050 +         DEFB RELAB1+1
              1050 +         IF   RELAB1.EQ.INTPC
              1050 +         ENDIF
1528 00       1051           DEFB 0
1529 EB       1052           EX   DE,HL
152A 0600     1053           LD   B,0
152C 110104   1054           LD   DE,PATYSZ.SHL.8+PATXSZ
152F          1055           SYSTEM BLANK
152F FF       1055 +         RST  56
1530 2A       1055 +         DEFB BLANK
              1055 +         IF   BLANK.EQ.INTPC
              1055 +         ENDIF
1531 D1       1056           POP  DE
1532 C9       1057           RET
1533          1058  CRASH:
              1059           ;A PLAYER HAS CRASHED.  DESTROY HIS ARROW AND ELIM
              1060           ;HIM FROM THE GAME.
1533 016D17   1061           LD   BC,EXPATS
1536 118117   1062           LD   DE,EXCOLS      ;DE<-EXPLODE COLOR TABLE ADDR
1539 3E05     1063           LD   A,5
153B 21B217   1064           LD   HL,EXPSND
153E F5       1065  EXLOOP:  PUSH AF            ;PUSH LOOP COUNT
153F C5       1066           PUSH BC            ;PUSH EXT PAT ADDR
1540 D5       1067           PUSH DE            ;PUSH EXPLODE COLOR TBL ADDR
1541 E5       1068           PUSH HL            ;PUSH EXPLODE SOUND ADDR
1542 1A       1069           LD   A,(DE)        ;A<-EXPLODE COLOR
1543 D300     1070           OUT  (COLOR),A
1545 C5       1071           PUSH BC
1546 DD5605   1072           LD   D,(IX+ARRY)
1549 DD5E04   1073           LD   E,(IX+ARRX)
154C CD2515   1074           CALL ERASE
154F E1       1075           POP  HL            ;PAT ADDR
1550 3E10     1076           LD   A,WRITOR
1552 010104   1077           LD   BC,PATYSZ.SHL.8+PATXSZ
1555          1078           SYSTEM WRIT        ;WRIT EXPLOSION
1555 FF       1078 +         RST  56
1556 24       1078 +         DEFB WRIT
              1078 +         IF   WRIT.EQ.INTPC
              1078 +         ENDIF
1557          1079           SYSSUK PAWS
1557 FF       1079 +         RST  56
1558 51       1079 +         DEFB PAWS+1
              1079 +         IF   PAWS.EQ.INTPC
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               1079 +         ENDIF
1559 07        1080           DEFB 7
155A E1        1081           POP  HL            ;GET EXPLODE SOUND ADDR
155B 011808    1082           LD   BC,SBLEN.SHL.8+SNDBX
155E EDB3      1083           OTIR
1560 D1        1084           POP  DE
1561 C1        1085           POP  BC
1562 F1        1086           POP  AF
1563 3D        1087           DEC  A
1564 2807      1088           JR   Z,EXPFIN-$    ; LOOP COUNT EXPIRED
1566 13        1089           INC  DE            ;INC TO NEXT COLOR
1567 03        1090           INC  BC            ;BUMP UP TO NEXT PAT ADDR
1568 03        1091           INC  BC
1569 03        1092           INC  BC
156A 03        1093           INC  BC
156B 18D1      1094           JR   EXLOOP-$
156D           1095  EXPFIN:
156D FD5605    1096           LD   D,(IY+PPATH)
1570 FD5E04    1097           LD   E,(IY+PPATL) ;DE<-PLAYER PAT ADDR
1573 FD210000  1098           LD   IY,0
1577 FD19      1099           ADD  IY,DE         ;IY<-PLAYER PAT ADDR
1579 110004    1100           LD   DE,4.SHL.8+0  ;D<-LOOP COUNT
157C FD7E00    1101  STOMP:   LD   A,(IY+0)      ;A<-BYTE OF PLAYER PATTERN
157F 21B841    1102           LD   HL,STARTS
1582 01200D    1103           LD   BC,ALLBYT
1585 EDB1      1104  STLOOP:  CPIR
1587 2005      1105           JR   NZ,RESTOM-$
1589 03        1106           INC  BC
158A 2B        1107           DEC  HL
158B 73        1108           LD   (HL),E
158C 18F7      1109           JR   STLOOP-$
158E FD23      1110  RESTOM:  INC  IY
1590 15        1111           DEC  D
1591 20E9      1112           JR   NZ,STOMP-$
1593 DDCB0676  1113           BIT  HUMBIT,(IX+PSTAT) ;
1597 2804      1114           JR   Z,KILLST-$    ;IF HUMAN
1599 21A44F    1115           LD   HL,CNOHUM
159C 35        1116           DEC  (HL)          ;DEC CURRENT # HUMANS ENDIF
159D DDCB06BE  1117  KILLST:  RES  ACTBIT,(IX+PSTAT) ;KILL STATUS
               1118           ;INC ALL ACTIVE PLAYERS SCORES
15A1 0E04      1119           LD   C,4
15A3           1120  BUMPEM:
15A3 0D        1121           DEC  C
15A4 79        1122           LD   A,C
15A5 CD5C16    1123           CALL LDPLIX
15A8 DDCB067E  1124           BIT  ACTBIT,(IX+PSTAT)
15AC 2818      1125           JR   Z,BUMPCK-$
15AE 0600      1126           LD   B,0
15B0 C5        1127           PUSH BC
15B1 79        1128           LD   A,C
15B2 21964F    1129           LD   HL,CURSCR
15B5 09        1130           ADD  HL,BC
15B6 09        1131           ADD  HL,BC
15B7 09        1132           ADD  HL,BC
15B8 37        1133           SCF
15B9 CDE315    1134           CALL DISPSC
15BC C1        1135           POP  BC
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

15BD          1136          SYSTEM INCSCR
15BD FF       1136 +        RST  56
15BE 54       1136 +        DEFB INCSCR
              1136 +        IF   INCSCR.EQ.INTPC
              1136 +        ENDIF
15BF 79       1137          LD   A,C
15C0 C5       1138          PUSH BC
15C1 B7       1139          OR   A            ;RESET CARRY
15C2 CDE315   1140          CALL DISPSC
15C5 C1       1141          POP  BC
15C6          1142  BUMPCK:
15C6          1143          SYSSUK PAWS
15C6 FF       1143 +        RST  56
15C7 51       1143 +        DEFB PAWS+1
              1143 +        IF   PAWS.EQ.INTPC
              1143 +        ENDIF
15C8 1E       1144          DEFB 30
15C9 79       1145          LD   A,C
15CA B7       1146          OR   A
15CB 20D6     1147          JR   NZ,BUMPEM-$
              1148          ;DEC CURR # PLAYERS
              1149          ;IF CURR # PLAYERS LEQ 1 GO TO END GAME
15CD 21A24F   1150          LD   HL,CNOPL
15D0 35       1151          DEC  (HL)
15D1 35       1152          DEC  (HL)
15D2 2802     1153          JR   Z,ENDCHK-$
15D4 34       1154          INC  (HL)
15D5 C9       1155          RET
15D6          1156  ENDCHK:
15D6 3ADC4F   1157          LD   A,(CT7)
15D9 3D       1158          DEC  A
15DA 27       1159          DAA
15DB 32DC4F   1160          LD   (CT7),A
15DE C24013   1161          JP   NZ,FIREIT
15E1          1162          SYSTEM QUIT
15E1 FF       1162 +        RST  56
15E2 78       1162 +        DEFB QUIT
              1162 +        IF   QUIT.EQ.INTPC
              1162 +        ENDIF
15E3          1163  DISPSC:
              1164          ;DISPLAY SCORE
              1165          ;A=PLAYER#
              1166          ;HL->LAST BYTE OF SCORE
15E3 FD4E09   1167          LD   C,(IY+PSDOP)
15E6 3004     1168          JR   NC,NOTXOR-$
15E8 CBA1     1169          RES  MROR,C
15EA CBE9     1170          SET  MRXOR,C
15EC          1171  NOTXOR:
15EC FD5E07   1172          LD   E,(IY+PSPOSX)
15EF FD5608   1173          LD   D,(IY+PSPOSY)
15F2 3E0C     1174          LD   A,12
15F4 83       1175          ADD  A,E
15F5 5F       1176          LD   E,A
15F6 14       1177          INC  D
15F7 0643     1178          LD   B,43H
15F9 DD210D02 1179          LD   IX,FNTSML
15FD          1180          SYSTEM DISNUM
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND          COMMENT

15FD FF       1180 +         RST  56
15FE 36       1180 +         DEFB DISNUM
              1180 +         IF   DISNUM.EQ.INTPC
              1180 +         ENDIF
15FF C9       1181           RET
1600          1182  ANIARR:
              1183           ;ANIMATE THE ARROW
              1184           ;INPUT AND OUTPUT IS IX WHO STAYS THE SAME
              1185           ;DESTROYS ALL OTHER REGISTERS
1600 DDCB067E 1186           BIT  ACTBIT,(IX+PSTAT)
1604 C8       1187           RET  Z                ;EXIT IF NOT ACTIVE
1605 DD7E03   1188           LD   A,(IX+AROT)
1608 CD2F17   1189           CALL GETROT           ;HL<-ARROW PAT ADDR
160B DD5605   1190           LD   D,(IX+ARRY)
160E DD5E04   1191           LD   E,(IX+ARRX)
1611 E5       1192           PUSH HL
1612 CD2515   1193           CALL ERASE
1615 E1       1194           POP  HL
1616 010104   1195           LD   BC,PATYSZ.SHL.8+PATXSZ
1619 3E10     1196           LD   A,WRITOR
161B          1197           SYSTEM WRIT
161B FF       1197 +         RST  56
161C 24       1197 +         DEFB WRIT
              1197 +         IF   WRIT.EQ.INTPC
              1197 +         ENDIF
161D C9       1198           RET
161E          1199  STALL:
              1200           ;THIS ROUTINE TAKES CARE OF ARROW ANIMATION
              1201           ;AND SHOWING A PLAYER HIS CURRENT JOY STICK POSIT
              1202           ;A=WHICH PLAYER:
              1203           ;B=JOY STICK BITS
161E CD5C16   1204           CALL LDPLIX           ;IX<-ADDR OF PLAYER PACKET
1621 AF       1205           XOR  A
1622 B0       1206           OR   B
1623 2003     1207           JR   NZ,STORIT-$
1625 DD7E02   1208           LD   A,(IX+CURSW)
1628 DD7702   1209  STORIT:  LD   (IX+CURSW),A
162B DDAE01   1210           XOR  (IX+LASTMV)      ;A<-DIFFERENCE FROM LAST MOVE
162E 2812     1211           JR   Z,GETLM-$       ;IF DIFFERENCE=0 USE LAST MOVE
1630 EE0C     1212           XOR  RLMOVE
1632 280E     1213           JR   Z,GETLM-$
1634 EE0C     1214           XOR  RLMOVE
1636 EE03     1215           XOR  UDMOVE
1638 2808     1216           JR   Z,GETLM-$
163A EE03     1217           XOR  UDMOVE
163C DDCB0676 1218  HUMCHK:  BIT  HUMBIT,(IX+PSTAT)
1640 2003     1219           JR   NZ,GOTIT-$      ;IF HUMAN WE'VE GOT IT
1642 DD7E01   1220  GETLM:   LD   A,(IX+LASTMV) ;GET LAST MOVE
1645 DD7703   1221  GOTIT:   LD   (IX+AROT),A   ;STORE ARROW ROTATION
1648 18B6     1222           JR   ANIARR-$
164A          1223  TICKIT:
              1224           ;TICK COUNT<-(8-CURR # PLAYERS)
164A 3AA44F   1225           LD   A,(CNOHUM)
164D B7       1226           OR   A
164E 3E02     1227           LD   A,2
1650 2806     1228           JR   Z,STICK-$
1652 21A24F   1229           LD   HL,CNOPL
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1655 3E08      1230           LD   A,8
1657 96        1231           SUB  (HL)
1658 32D54F    1232  STICK:   LD   (CTO),A
165B C9        1233           RET
165C           1234  LDPLIX:
165C           1235  LDPLIY:
               1236           ;LOAD IY WITH POINTER TO CURR PLAYER ROM DATA
               1237           ;LOAD IX WITH A POINTER TO CURRENT PLAYER PACKET
               1238           ;A=PLAYER# MUST BE GEQ 0 & LEQ 3
165C D5        1239           PUSH DE
165D E5        1240           PUSH HL
165E           1241           SYSSUK INDEXW
165E FF        1241 +         RST  56
165F 5B        1241 +         DEFB INDEXW+1
               1241 +         IF   INDEXW.EQ.INTPC
               1241 +         ENDIF
1660 6F16      1242           DEFW ROMTBL
1662 D5        1243           PUSH DE
1663 FDE1      1244           POP  IY
1665           1245           SYSSUK INDEXW
1665 FF        1245 +         RST  56
1666 5B        1245 +         DEFB INDEXW+1
               1245 +         IF   INDEXW.EQ.INTPC
               1245 +         ENDIF
1667 7716      1246           DEFW RAMTBL
1669 D5        1247           PUSH DE
166A DDE1      1248           POP  IX
166C E1        1249           POP  HL
166D D1        1250           POP  DE
166E C9        1251           RET
166F 4517      1252  ROMTBL:  DEFW PLROM0
1671 4F17      1253           DEFW PLROM1
1673 5917      1254           DEFW PLROM2
1675 6317      1255           DEFW PLROM3
1677 A84F      1256  RAMTBL:  DEFW PLAY0
1679 AF4F      1257           DEFW PLAY1
167B B64F      1258           DEFW PLAY2
167D BD4F      1259           DEFW PLAY3
167F           1260  RANMOV:
               1261           ;GENERATE A RANDOM MOVE FOR THE PLAYER PACKET POI
               1262           ;INPUT AND OUTPUT:
               1263           ;B=CURRENT SWITCH C=LAST SWITCH
167F           1264           SYSSUK RANGED
167F FF        1264 +         RST  56
1680 77        1264 +         DEFB RANGED+1
               1264 +         IF   RANGED.EQ.INTPC
               1264 +         ENDIF
1681 20        1265           DEFB 32
1682 B7        1266           OR   A              ;TIME TO CHANGE DIRECTION?
1683 2808      1267           JR   Z,NEWMOV-$
1685 DD4601    1268           LD   B,(IX+LASTMV) ;USE LAST MOVE
1688 78        1269           LD   A,B
1689 CDAC16    1270           CALL MOVTST
168C C8        1271           RET  Z              ;LAST MOVE IS GOOD ENOUGH
168D           1272  NEWMOV:  SYSSUK RANGED
168D FF        1272 +         RST  56
168E 77        1272 +         DEFB RANGED+1
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

              1272 +         IF   RANGED.EQ.INTPC
              1272 +         ENDIF
168F 04       1273           DEFB 4
1690 47       1274           LD   B,A
1691 04       1275           INC  B
1692 3E80     1276           LD   A,80H
1694 07       1277  SHFTIT:  RLCA
1695 10FD     1278           DJNZ SHFTIT-$
1697 47       1279           LD   B,A
1698 3E08     1280  RANFIN:  LD   A,08H
169A CDAC16   1281           CALL MOVTST
169D 2002     1282           JR   NZ,ANYMOV-$
169F 47       1283           LD   B,A
16A0 C9       1284           RET
16A1 060F     1285  ANYMOV:  LD   B,0FH          ;TRY ALL MOVES
16A3 3E08     1286           LD   A,08H
16A5 CDAC16   1287           CALL MOVTST
16A8 47       1288           LD   B,A
16A9 C9       1289           RET
16AA          1290  MOVANY:
16AA 3E08     1291           LD   A,AMOVE
16AC          1292  MOVTST:
              1293           ;TEST THE NEW MOVE FOR VALIDITY
              1294           ;THE INPUTS AND OUTPUTS.
              1295           ;B=A SET OF MOVES TO BE TESTED (IS UNCHANGED)
              1296           ;C=UNCHANGED
              1297           ;A=FIRST MOVE TO TEST,VALUE OF GOOD MOVE ON OUTPU
              1298           ;DE,HL=RETURNED UNCHANGED
              1299           ;D=# ROTATES
              1300           ;Z FLAG=Z IF GOOD MOVE FOUND(A CONTAINS FIRST GOO
              1301           ;Z FLAG=NZ IF NO GOOD MOVES FOUND(IN B)
16AC D5       1302           PUSH DE
16AD 1608     1303           LD   D,8            ;INIT # ROTATES
16AF 0F       1304  ROTMSK:  RRCA                ;ROTATE TO NEXT MOVE
16B0 5F       1305           LD   E,A
16B1 A0       1306           AND  B
16B2 CDC016   1307           CALL CHKMOV         ;CHECK MOVE
16B5 7B       1308           LD   A,E
16B6 2806     1309           JR   Z,MOVEXT-$     ;FOUND ONE
16B8 15       1310           DEC  D              ;DEC # ROTATES
16B9 20F4     1311           JR   NZ,ROTMSK-$
16BB 37       1312           SCF                 ;NO GOOD MOVES
16BC CB12     1313           RL   D              ;SET Z FLAG=NZ
16BE D1       1314  MOVEXT:  POP  DE
16BF C9       1315           RET
16C0          1316  CHKMOV:
              1317           ;CHECK THE MOVE IN A FOR BEING UNOCCUPIED
              1318           ;INPUT AND OUTPUT:
              1319           ;A=UP,DOWN,RIGHT OR LEFT BIT (RETURNED UNCHANGED)
              1320           ;Z FLAG=Z IF MOVE IN A IS TO AN EMPTY POSITION
              1321           ;Z FLAG=NZ IF MOVE IN A IS BAD
              1322           ;BC,DE,HL RETURNED UNTOUCHED
              1323           ;IX=POINTER TO CURRENT PLAYER PACKET
              1324           ;LOCAL TO THIS ROUTINE:
              1325           ;    D=TEMP X COORD OF ARROW
              1326           ;    E=TEMP Y COORD OF ARROW
16C0 C5       1327           PUSH BC
```

```
ADDR OBJECT    STMT   LABEL    OPCD OPERAND        COMMENT

16C1 D5        1328            PUSH DE
16C2 E5        1329            PUSH HL
16C3 F5        1330            PUSH AF
16C4 DD5604    1331            LD   D,(IX+ARRX)  ;GET X COORD OF ARROW
16C7 DD5E05    1332            LD   E,(IX+ARRY)  ;GET Y COORD OF ARROW
16CA CB57      1333   TLEFT:   BIT  CHLEFT,A
16CC 280A      1334            JR   Z,TRIGHT-$
16CE 7A        1335            LD   A,D          ;GOT A  LEFT MOVE
16CF FE00      1336            CP   LOWX
16D1 282F      1337            JR   Z,BADMOV-$   ;ALREADY AT LOWEST X
16D3 D604      1338            SUB  WIDTH        ;DEC TEMP X BY 1 POSITION
16D5 57        1339            LD   D,A
16D6 1830      1340            JR   LOOKSQ-$
16D8 CB5F      1341   TRIGHT:  BIT  CHRIGH,A
16DA 280A      1342            JR   Z,TUP-$
16DC 7A        1343            LD   A,D          ;GOT A RIGHT MOVE
16DD FE9C      1344            CP   XMAX
16DF 3021      1345            JR   NC,BADMOV-$  ;ALREADY GEQ MAX X
16E1 C604      1346            ADD  A,WIDTH
16E3 57        1347            LD   D,A
16E4 1822      1348            JR   LOOKSQ-$
16E6 CB47      1349   TUP:     BIT  CHUP,A
16E8 280A      1350            JR   Z,TDOWN-$
16EA 7B        1351            LD   A,E          ;GOT AN UP MOVE
16EB FE0B      1352            CP   LOWY
16ED 2813      1353            JR   Z,BADMOV-$   ;ALREADY AT LOWEST Y
16EF D604      1354            SUB  HEIGHT       ;DEC TEMP Y BY 1 POSITION
16F1 5F        1355            LD   E,A
16F2 1814      1356            JR   LOOKSQ-$
16F4 CB4F      1357   TDOWN:   BIT  CHDOWN,A
16F6 280A      1358            JR   Z,BADMOV-$
16F8 7B        1359            LD   A,E          ;GOT A DOWN MOVE
16F9 FE5B      1360            CP   YMAX
16FB 2805      1361            JR   Z,BADMOV-$   ;ALREADY AT HIGHEST Y
16FD C604      1362            ADD  A,HEIGHT     ; INC TEMP Y BY 1 POSITION
16FF 5F        1363            LD   E,A
1700 1806      1364            JR   LOOKSQ-$
1702 F1        1365   BADMOV:  POP  AF
1703 37        1366            SCF
1704 CB12      1367            RL   D            ;SET Z FLAG = NZ
1706 1823      1368            JR   MOVEND-$
1708           1369   LOOKSQ:
               1370            ;SEE IF THE NEW SQUARE IS OCCUPIED
1708 D5        1371            PUSH DE
1709 D5        1372            PUSH DE
170A C1        1373            POP  BC
170B 51        1374            LD   D,C          ;REVERSE X,Y FOR SYSTEM
170C 58        1375            LD   E,B
170D           1376            SYSSUK RELAB1
170D FF        1376 +          RST  56
170E 3B        1376 +          DEFB RELAB1+1
               1376 +          IF   RELAB1.EQ.INTPC
               1376 +          ENDIF
170F 00        1377            DEFB 0
1710 E1        1378            POP  HL
1711 EB        1379            EX   DE,HL
1712 7E        1380            LD   A,(HL)
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1713 B7        1381           OR   A             ;TEST SQUARE
1714 20EC      1382           JR   NZ,BADMOV-$
1716 012800    1383           LD   BC,BYTEPL
1719 09        1384           ADD  HL,BC
171A 7E        1385           LD   A,(HL)
171B B7        1386           OR   A
171C 20E4      1387           JR   NZ,BADMOV-$
171E 7A        1388           LD   A,D
171F 32A54F    1389           LD   (TARRX),A     ;STORE TEMP ARROW X COORD
1722 7B        1390           LD   A,E
1723 32A64F    1391           LD   (TARRY),A     ;STORE TEMP ARROW Y COORD
1726 F1        1392           POP  AF
1727 1600      1393           LD   D,0
1729 CB3A      1394           SRL  D             ;SET Z FLAG=Z
172B E1        1395  MOVEND:  POP  HL
172C D1        1396           POP  DE
172D C1        1397           POP  BC
172E C9        1398           RET
172F           1399  GETROT:
               1400           ;HL<-BASE ADDR OF ROTATED PATTERN
               1401           ;A<-DIRECTION OF ROTATION
               1402           ;IF A HAS MORE THAN 1 BIT SET THEN ONLY ONE IS US
172F 218A17    1403           LD   HL,AUP
1732 CB47      1404           BIT  CHUP,A
1734 C0        1405           RET  NZ
1735 219217    1406           LD   HL,ADOWN
1738 CB4F      1407           BIT  CHDOWN,A
173A C0        1408           RET  NZ
173B 218E17    1409           LD   HL,ARIGHT
173E CB5F      1410           BIT  CHRIGH,A
1740 C0        1411           RET  NZ
1741 219617    1412           LD   HL,ALEFT
1744 C9        1413           RET
               1414           ;START OF ROM DATA FOR EACH PLAYER.
               1415           ;CONTAINS: 4 PLAYER NOTES, PLAYER PATTERN ADDR
               1416           ;  , PLAYER CHAR DISP OPT
               1417           ;PLAYER SCORE DISP OPT
               1418           ;AND PLAYER SCORE POSITION
1745           1419  PLROM0:
1745           1420  PNOTE0:  DEF4X G0,GS0,A0,AS0
1745 FD        1420 +         DEFB G0
1746 EE        1420 +         DEFB GS0
1747 E1        1420 +         DEFB A0
1748 D4        1420 +         DEFB AS0
1749 9A17      1421  PPADR0:  DEFW PPAT0
174B 18        1422  PCDOP0:  DEFB 011000B
174C 04        1423  PSPOS0:  DEFB 4
174D 01        1424           DEFB 1
174E 18        1425  PSDOP0:  DEFB 011000B
174F           1426  PLROM1:
174F           1427  PNOTE1:  DEF4X B0,C1,CS1,D1
174F C8        1427 +         DEFB B0
1750 BD        1427 +         DEFB C1
1751 B2        1427 +         DEFB CS1
1752 A8        1427 +         DEFB D1
1753 9E17      1428  PPADR1:  DEFW PPAT1
1755 1C        1429  PCDOP1:  DEFB 011100B
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1756 85        1430  PSPOS1:  DEFB 133
1757 01        1431           DEFB 1
1758 1C        1432  PSDOP1:  DEFB 011100B
1759           1433  PLROM2:
1759           1434  PNOTE2:  DEF4X DS1,E1,F1,FS1
1759 9F        1434 +         DEFB DS1
175A 96        1434 +         DEFB E1
175B 8D        1434 +         DEFB F1
175C 85        1434 +         DEFB FS1
175D A217      1435  PPADR2:  DEFW PPAT2
175F 1C        1436  PCDOP2:  DEFB 011100B
1760 2D01      1437  PSPOS2:  DEFW 45+1.SHL.8
1762 1C        1438  PSDOP2:  DEFB 011100B
1763           1439  PLROM3:
1763           1440  PNOTE3:  DEF4X G1,GS1,A1,AS1
1763 7E        1440 +         DEFB G1
1764 77        1440 +         DEFB GS1
1765 70        1440 +         DEFB A1
1766 6A        1440 +         DEFB AS1
1767 A617      1441  PPADR3:  DEFW PPAT3
1769 18        1442  PCDOP3:  DEFB 011000B
176A 5D01      1443  PSPOS3:  DEFW 93+1.SHL.8
176C 18        1444  PSDOP3:  DEFB 011000B
               1445           ;EXPLOSION PATTERNS
176D           1446  EXPATS:
176D           1447  EXPAT1:  DEF4X 0,00010100B,00010100B,0
176D 00        1447 +         DEFB 0
176E 14        1447 +         DEFB 00010100B
176F 14        1447 +         DEFB 00010100B
1770 00        1447 +         DEFB 0
1771           1448  EXPAT2:  DEF4X 0,01000101B,01010001B,0
1771 00        1448 +         DEFB 0
1772 45        1448 +         DEFB 01000101B
1773 51        1448 +         DEFB 01010001B
1774 00        1448 +         DEFB 0
1775           1449  EXPAT3:  DEF4X 00000101B,01000000B,00000001B,01010000B
1775 05        1449 +         DEFB 00000101B
1776 40        1449 +         DEFB 01000000B
1777 01        1449 +         DEFB 00000001B
1778 50        1449 +         DEFB 01010000B
1779           1450  EXPAT4:  DEF4X 00010001B,01000000B,00000001B,01000100B
1779 11        1450 +         DEFB 00010001B
177A 40        1450 +         DEFB 01000000B
177B 01        1450 +         DEFB 00000001B
177C 44        1450 +         DEFB 01000100B
177D           1451  EXPAT5:  DEF4X 0,0,0,0
177D 00        1451 +         DEFB 0
177E 00        1451 +         DEFB 0
177F 00        1451 +         DEFB 0
1780 00        1451 +         DEFB 0
               1452           ;EXPLOSION COLORS
1781           1453  EXCOLS:
1781 07        1454           DEFB 7
1782 03        1455           DEFB 3
1783 07        1456           DEFB 7
1784 03        1457           DEFB 3
1785 77        1458           DEFB 077H
```

```
ADDR OBJECT    STMT   LABEL    OPCD OPERAND         COMMENT

               1459            ;COUNT DOWN DISPLAY PACKET
1786 0400      1460   CDCOLR:  DEFW 0100B+0.SHL.8
               1461            ;TIMER DISPLAY PACKET
1788 0180      1462   TDPACK:  DEFW 0001B+10000000B.SHL.8
               1463            ;ARROW ANIMATION PATTERNS FOR EACH ROTATION
178A           1464   AUP:
178A           1465            DEF4X 00010100B,01010101B,01000001B,0
178A 14        1465 +          DEFB 00010100B
178B 55        1465 +          DEFB 01010101B
178C 41        1465 +          DEFB 01000001B
178D 00        1465 +          DEFB 0
178E           1466   ARIGHT:
178E           1467            DEF4X 00010100B,00000101B,00000101B,00010100B
178E 14        1467 +          DEFB 00010100B
178F 05        1467 +          DEFB 00000101B
1790 05        1467 +          DEFB 00000101B
1791 14        1467 +          DEFB 00010100B
1792           1468   ADOWN:
1792           1469            DEF4X 0,01000001B,01010101B,00010100B
1792 00        1469 +          DEFB 0
1793 41        1469 +          DEFB 01000001B
1794 55        1469 +          DEFB 01010101B
1795 14        1469 +          DEFB 00010100B
1796           1470   ALEFT:
1796           1471            DEF4X 00010100B,01010000B,01010000B,00010100B
1796 14        1471 +          DEFB 00010100B
1797 50        1471 +          DEFB 01010000B
1798 50        1471 +          DEFB 01010000B
1799 14        1471 +          DEFB 00010100B
               1472            ;PLAYER PATTERNS
179A           1473   PPAT0:   DEF4X 00001000B,10101000B,00101010B,00100000B
179A 08        1473 +          DEFB 00001000B
179B A8        1473 +          DEFB 10101000B
179C 2A        1473 +          DEFB 00101010B
179D 20        1473 +          DEFB 00100000B
179E           1474   PPAT1:   DEF4X 11111111B,11000011B,11000011B,11111111B
179E FF        1474 +          DEFB 11111111B
179F C3        1474 +          DEFB 11000011B
17A0 C3        1474 +          DEFB 11000011B
17A1 FF        1474 +          DEFB 11111111B
17A2           1475   PPAT2:   DEF4X 00001100B,11111100B,00111111B,00110000B
17A2 0C        1475 +          DEFB 00001100B
17A3 FC        1475 +          DEFB 11111100B
17A4 3F        1475 +          DEFB 00111111B
17A5 30        1475 +          DEFB 00110000B
17A6           1476   PPAT3:   DEF4X 10101010B,10000010B,10000010B,10101010B
17A6 AA        1476 +          DEFB 10101010B
17A7 82        1476 +          DEFB 10000010B
17A8 82        1476 +          DEFB 10000010B
17A9 AA        1476 +          DEFB 10101010B
               1477            ;COLOR BLOCK
17AA           1478   CBLOCK:
17AA F8        1479            DEFB 0F8H
17AB F8        1480            DEFB 0F8H
17AC F8        1481            DEFB 0F8H
17AD F8        1482            DEFB 0F8H
17AE B5        1483            DEFB 0B5H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

17AF 52        1484           DEFB 052H
17B0 F8        1485           DEFB 0F8H
17B1 77        1486           DEFB 077H
               1487           ;EXPLOSION SOUNDS
17B2           1488  EXPSND:  DEF8 0EFH,0FFH,3FH,0,0FFH,0FDH,0F5H,0F5H
17B2 EF        1488 +         DEFB 0EFH
17B3 FF        1488 +         DEFB 0FFH
17B4 3F        1488 +         DEFB 3FH
17B5 00        1488 +         DEFB 0
17B6 FF        1488 +         DEFB 0FFH
17B7 FD        1488 +         DEFB 0FDH
17B8 F5        1488 +         DEFB 0F5H
17B9 F5        1488 +         DEFB 0F5H
17BA           1489           DEF8 08FH,0EEH,3EH,0,0FFH,0FDH,0F5H,0F5H
17BA 8F        1489 +         DEFB 08FH
17BB EE        1489 +         DEFB 0EEH
17BC 3E        1489 +         DEFB 3EH
17BD 00        1489 +         DEFB 0
17BE FF        1489 +         DEFB 0FFH
17BF FD        1489 +         DEFB 0FDH
17C0 F5        1489 +         DEFB 0F5H
17C1 F5        1489 +         DEFB 0F5H
17C2           1490           DEF8 04EH,088H,38H,0,0FFH,0FDH,0F5H,0F5H
17C2 4E        1490 +         DEFB 04EH
17C3 88        1490 +         DEFB 088H
17C4 38        1490 +         DEFB 38H
17C5 00        1490 +         DEFB 0
17C6 FF        1490 +         DEFB 0FFH
17C7 FD        1490 +         DEFB 0FDH
17C8 F5        1490 +         DEFB 0F5H
17C9 F5        1490 +         DEFB 0F5H
17CA           1491           DEF8 048H,044H,34H,0,0FFH,0FDH,0F5H,0F5H
17CA 48        1491 +         DEFB 048H
17CB 44        1491 +         DEFB 044H
17CC 34        1491 +         DEFB 34H
17CD 00        1491 +         DEFB 0
17CE FF        1491 +         DEFB 0FFH
17CF FD        1491 +         DEFB 0FDH
17D0 F5        1491 +         DEFB 0F5H
17D1 F5        1491 +         DEFB 0F5H
17D2           1492           DEF8 0,0,0,0,0,0,0,0
17D2 00        1492 +         DEFB 0
17D3 00        1492 +         DEFB 0
17D4 00        1492 +         DEFB 0
17D5 00        1492 +         DEFB 0
17D6 00        1492 +         DEFB 0
17D7 00        1492 +         DEFB 0
17D8 00        1492 +         DEFB 0
17D9 00        1492 +         DEFB 0
17DA           1493           END


TOTAL ASSEMBLER ERRORS =
                              CROSS REFERENCE

LABEL       VALUE        REFERENCE

A0           00E1         -508  1421
A1           0070         -520  1441
A2           0037         -532
A3           001B         -544
A4           000D         -556
A5           0006         -562
ACTBIT       0007         -694   957  1117  1124  1186
ACTINT       000E         -225   811
ACTION       146C         -893   943
ACTIVE       0080         -692   900   905
ADOWN        1792        -1368  1406
ALEFT        1796        -1369  1412
ALKEYS       0214          -49   938
ALLBYT       0D20         -719   991  1103
ALLHUM       13B7         -821   844
AMOVE        0008         -678  1291
ANIARR       1600        -1105  1046  1222
ANIMAX       0003         -672
ANYMOV       16A1        -1198  1282
ARIGHT       178E        -1367  1409
AROT         0003         -687   970  1033  1188  1221
ARRX         0004         -688   814   816   818   820   862  1035  1043
                          1073  1191  1331
ARRY         0005         -689   863  1034  1045  1072  1190  1332
AS0          00D4         -509  1421
AS1          006A         -521  1441
AS2          0034         -533
AS3          001A         -545
AUP          178A        -1366  1403
B0           00C8         -510  1428
B1           0064         -522
B2           0031         -534
B3           0018         -546
BADMOV       1702        -1278  1337  1345  1353  1358  1361  1382  1387
BCDADD       0062         -277
BCDCHS       006A         -281
BCDDIV       0068         -280
BCDMUL       0066         -279
BCDNEG       006C         -282
BCDSUB       0064         -278
BEGRAM       4FCE         -594   753
BITSPL       00A0          -43
BLANK        002A         -243  1056  1056
BMUSIC       0012         -229
BUMPCK       15C6        -1071  1125
BUMPEM       15A3        -1051  1147
BYTEPL       0028          -42   673   697   719   720  1383
C1           00BD         -511  1428
C2           005E         -523
C3           002E         -535
C4           0017         -547
C5           000B         -557
C6           0005         -563
```

```
C7          0002          -566
CALPIZ      148B          -909    948
CBA         0009          -123
CBB         0007          -121
CBC         0006          -120
CBD         0005          -119
CBE         0004          -118
CBFLAG      0008          -122
CBH         000B          -125
CBIXH       0003          -117
CBIXL       0002          -116
CBIYH       0001          -115
CBIYL       0000          -114
CBL         000A          -124
CBLEN       0008          -725
CBLOCK      17AA         -1372    777
CDCOLR      1786         -1362
CDOPT       0044          -681    923
CDOWNL      1426          -882    930
CHDOWN      0001          -111   1357    1407
CHKMOV      16CD         -1229   1307
CHLEFT      0002          -110   1333
CHRDIS      0032          -248    868     868     872     872     925     925
CHRIGH      0003          -109   1341    1410
CHTRIG      0004          -108
CHUP        0000          -112   1349    1404
CKNOPL      141E          -876    903
CKSUM3      1418          -873
CLEARF      14B4          -930    822     931
CNOHUM      4FA4          -743    838     851    1115    1225
CNOPL       4FA2          -741    786     787     846     853    1150    1229
CNT         4FDD          -611    935
COL0L       0004          -168
COLOR       0000          -164   1070
COL1L       0005          -169
COL1R       0001          -165
COL2L       0006          -170
COL2R       0002          -166
COL3L       0007          -171
COL3R       0003          -167
COLBX       000B          -172
COLLST      4FE8          -622
COLSET      0018          -234    777
CONCM       0008          -189
CRASH       1533          -993   1029
CS1         00B2          -512   1428
CS2         0059          -524
CS3         002C          -536
CS4         0015          -548
CS5         000A          -558
CT0         4FD5          -602   1232
CT1         4FD6          -603
CT2         4FD7          -604
CT3         4FD8          -605
CT4         4FD9          -606
CT5         4FDA          -607
CT6         4FDB          -608
```

```
CT7       4FDC         -609   761   830  1157  1160
CTIMER    0203          -46
CURSCR    4F96         -739   767   889  1129
CURSW     0002         -686   999  1208  1209
D1        00A8         -513  1428
D2        0054         -525
D3        0029         -537
D4        0014         -549
DABS      0072         -285
DADD      006E         -283
DECCTS    0010         -226
DISNUM    0036         -250   826   826  1181  1181
DISPSC    15E3        -1088   893  1134  1140
DISTIM    0052         -267
DOIT      0044         -260   940   940
DOITB     0046         -261
DONTD     13A1         -806
DS1       009F         -514  1435
DS2       004F         -526
DS3       0027         -538
DS4       0013         -550
DS5       0009         -559
DS6       0004         -564
DSMG      0070         -284
DURAT     4FEA         -624
E1        0096         -515  1435
E2        004A         -527
E3        0025         -539
E4        0012         -551
EMUSIC    0014         -230   779   928   928
END       00C0         -379   948   948
ENDCHK    15D6        -1083  1153
ENDRAM    4FC4         -752   753
ENDSCR    4FF4         -632
ERASE     1525         -987  1036  1074  1193
EXCOLS    1781        -1355  1062
EXLOOP    153E        -1000  1094
EXPAT1    176D        -1354
EXPAT2    1771        -1354
EXPAT3    1775        -1354
EXPAT4    1779        -1354
EXPAT5    177D        -1354
EXPATS    176D        -1353  1061
EXPFIN    156D        -1026  1088
EXPSND    17B2        -1382  1064
F1        008D         -516  1435
F2        0046         -528
F3        0022         -540
F4        0011         -552
F5        0008         -560
FILL      001A         -235   767   767   781   786   990   990
FIREIT    1340         -764  1161
FIRSTC    2000          -40
FNTSML    020D          -48   824  1179
FNTSYS    0206          -47
FORCEM    00F6         -716
FPLAY     13B5         -820   842
```

```
FS1       0085        -517   1435
FS2       0042        -529
FS3       0020        -541
FS4       0010        -553
FTBASE    0000         -93
FTBYTE    0003         -96
FTFSX     0001         -94
FTFSY     0002         -95
FTPTH     0006         -99
FTPTL     0005         -98
FTYSIZ    0004         -97
G0        00FD        -506   1421
G1        007E        -518   1441
G2        003E        -530
G3        001F        -542
G4        000F        -554
G5        0007        -561
G6        0003        -565
G7        0001        -567
G8        0000        -568
GAMSTB    4FF8        -634
GETLM     1642       -1141   1211   1213   1216
GETNUM    004E        -265
GETPAR    004C        -264    759    759    763    763
GETROT    172F       -1310   1189
GOTBIT    14A1        -919    974
GOTIT     1645       -1142   1219
GOTMOV    14F7        -971   1018   1021   1025
GOTNPL    13AA        -812    835
GOTSW     14DC        -956   1012
GS0       00EE        -507   1421
GS1       0077        -519   1441
GS2       003B        -531
GS3       001D        -543
GS4       000E        -555
GSBEND    0007         -62
GSBSCR    0001         -61
GSBTIM    0000         -60
GTMINS    4FEE        -628
GTPLIX    13C4        -831    911
GTSECS    4FED        -627
HEIGHT    0004        -718   1354   1362
HORAF     000F        -195
HORCB     0009        -173
HUMAN     0040        -693    900
HUMANR    0040        -257
HUMBIT    0006        -695   1000   1113   1218
HUMCHK    163C       -1139
INCIX     146F        -897    958
INCSCR    0054        -268   1137   1137
INDEXB    005C        -274
INDEXN    0056        -271
INDEXW    005A        -273   1242   1242   1246   1246
INFBK     000D        -186
INLIN     000F        -188
INMOD     000E        -187
INTIPP    13BA        -826
```

```
INTPC    0000     -216   759   763   767   775   775   775   826
                   868   872   883   916   925   926   928   938
                   940   967   990  1042  1051  1056  1079  1080
                  1137  1144  1163  1181  1198  1242  1246  1265
                  1273  1377
INTP@    0000     -768  -789
INTST    0008     -193
JUSJOY   000F     -724
KCTASC   0040     -258
KEY0     0014     -206
KEY1     0015     -207
KEY2     0016     -208
KEY3     0017     -209
KEYSEX   4FE3     -617
KILLST   159D    -1048  1114
LASTMV   0001     -685  1016  1026  1032  1210  1220  1268
LASTSW   0000     -684   998  1009
LDPLIX   165C    -1155   857   956  1123  1204
LDPLIY   165C    -1156   969
LOOKSQ   1708    -1282  1340  1348  1356  1364
LOOPY    144F     -896   941
LOWX     0000     -677  1336
LOWY     000B     -675   676   701   702   703   720   790  1352
MAGIC    000C     -190
MATH     0056     -270
MCALL    0006     -219
MENU     004A     -263
MENUST   0218      -50
MJUMP    000A     -221
MOVANY   16AA    -1203  1020  1024  1028
MOVE     005E     -275
MOVEIT   14BC     -935   959
MOVEND   172B    -1306  1368
MOVEXT   16BE    -1227  1309
MOVJOY   14B4     -905   944   945   946   947
MOVTST   16AC    -1205  1017  1270  1281  1287
MRET     0008     -220
MRFLOP   0006     -101
MRLOCK   4FF7     -633
MROR     0004     -103  1169
MRROT    0002     -105
MRSHFT   0003     -106
MRXOR    0005     -102  1170
MRXPND   0003     -104
MSKTD    007E     -291
MUSVOL   0009     -679   982
MUZAK    0012     -228
MUZPC    4FCE     -596
MUZSP    4FD0     -597
MXSCR    021E      -51
NEGT     0074     -286
NEWMOV   168D    -1187  1267
NEWWAY   0001     -666
NGBIT    0002     -670
NOCUR    14CB     -946  1001
NOGAME   0235      -53   759
NOPLAY   0228      -52   763
```

```
NORMEM    4000       -39    720    736
NOTE0     0000      -705
NOTE1     0001      -706
NOTE2     0002      -707
NOTE3     0003      -708
NOTHUM    1419      -874    899
NOTXOR    15EC     -1096   1168
NPBIT     0003      -671
NUMPLY    4FF3      -631    765    833
NWHDWR    0001       -36    665
OA1       008F      -576
OA2       0047      -577
OA3       0023      -578
OA4       0011      -579    984
OA5       0008      -580
OB0       00FE      -570
OC0       00F1      -571
OD1       00D6      -572
OE1       00BF      -573
OF1       00B4      -574
OG1       00A0      -575
OLDWAY    0000      -665    666
ONETIM    1328      -755
OPOT0     4FDF      -613
OPOT1     4FE0      -614
OPOT2     4FE1      -615
OPOT3     4FE2      -616
OSW0      4FE4      -618    781
OSW1      4FE5      -619
OSW2      4FE6      -620
OSW3      4FE7      -621
PATDIM    0104      -723
PATXSZ    0001      -721    723    878   1039   1054   1077   1195
PATYSZ    0004      -722    723    878   1039   1054   1077   1195
PAWS      0050      -266    916    916    926    926   1080   1080   1144
                    1144
PCDOP     0006      -711    866
PCDOP0    174B     -1332
PCDOP1    1755     -1338
PCDOP2    175F     -1344
PCDOP3    1769     -1349
PIZBRK    0048      -262    967    967
PLAY0     4FA8      -748    814   1256
PLAY1     4FAF      -749    816   1257
PLAY2     4FB6      -750    818   1258
PLAY3     4FBD      -751    787    820   1259
PLIX      4FA3      -742    917    952    955    968
PLROM0    1745     -1330   1252
PLROM1    174F     -1336   1253
PLROM2    1759     -1342   1254
PLROM3    1763     -1347   1255
PNOTE0    1745     -1331
PNOTE1    174F     -1337
PNOTE2    1759     -1343
PNOTE3    1763     -1348
POT0      001C      -201
POT1      001D      -202
```

```
POT2       001E     -203
POT3       001F     -204
PPACKS     4FA8     -747
PPADR0     1749    -1331
PPADR1     1753    -1337
PPADR2     175D    -1343
PPADR3     1767    -1348
PPAT0      179A    -1371   1421
PPAT1      179E    -1371   1428
PPAT2      17A2    -1371   1435
PPAT3      17A6    -1371   1441
PPATH      0005     -710    879   1037   1096
PPATL      0004     -709    880   1038   1097
PRIOR      4FF9     -635
PSDOP      0009     -714   1167
PSDOP0     174E    -1335
PSDOP1     1758    -1341
PSDOP2     1762    -1346
PSDOP3     176C    -1351
PSPOS0     174C    -1333
PSPOS1     1756    -1339
PSPOS2     1760    -1345
PSPOS3     176A    -1350
PSPOSX     0007     -712    868   1172
PSPOSY     0008     -713    869   1173
PSTAT      0006     -690    748    749    750    751    787    901    906
                     957   1000   1113   1117   1124   1186   1218
PSWCY      0000      -58
PSWPV      0002      -57
PSWSGN     0007      -55
PSWZRO     0006      -56
PVOLAB     4FD2     -598
PVOLMC     4FD3     -599
QUIT       0078     -288   1163   1163
RAMTBL     1677    -1173   1246
RANFIN     1698    -1193
RANGED     0076     -287   1265   1265   1273   1273
RANMOV     167F    -1177   1014
RANSHT     4FEF     -630
RANTST     14D0     -950   1007
RCALL      0004     -218
RECTAN     001C     -236    795    799    802    805    808
RELAB1     003A     -253   1051   1051   1377   1377
RELABS     0038     -252
RESTOM     158E    -1041   1105
RESTOR     002E     -245
RLMOVE     000C     -668   1212   1214
RMASK      4FA7     -746
ROMTBL     166F    -1169   1242
ROTMSK     16AF    -1217   1311
RSTART     4FA1     -753
SAVE       002C     -244
SBLEN      0008     -726   1082
SCHEDR     000C     -224
SCREEN     0000      -41
SCROLL     0030     -246
SCRSTR     0016     -232
```

```
SCT0        0001          -128   943
SCT1        0002          -129
SCT2        0003          -130
SCT3        0004          -131
SCT4        0005          -132
SCT5        0006          -133
SCT6        0007          -134
SCT7        0008          -135
SEMI4S      4FDE          -612
SENFLG      4FFA          -636
SENTRY      0042          -259   938     938
SETB        007A          -289
SETOUT      0016          -233   790
SETW        007C          -290
SF0         0009          -136
SF1         000A          -137
SF2         000B          -138
SF3         000C          -139
SF4         000D          -140
SF5         000E          -141
SF6         000F          -142
SF7         0010          -143
SHFTIT      1694         -1190  1278
SHIFTU      0060          -276
SJ0         0015          -152   944     961
SJ1         0017          -154   945
SJ2         0019          -156   946
SJ3         001B          -158   947
SKYD        0013          -145   948
SKYU        0012          -146
SNDBX       0018          -184  1082
SNUL        0000          -127
SP0         001C          -147
SP1         001D          -148
SP2         001E          -149
SP3         001F          -150
SSEC        0011          -144
ST0         0014          -151
ST1         0016          -153
ST2         0018          -155
ST3         001A          -157
STALL       161E         -1120   963
STARTS      41B8          -720   990    1102
STICK       1658         -1153  1228
STIMER      0200           -45
STLOOP      1585         -1035  1109
STOMP       157C         -1032  1112
STOREN      0058          -272
STORIT      1628         -1130  1207
STRDIS      0034          -249
SUCK        000C          -222
SW0         0010          -197
SW1         0011          -198
SW2         0012          -199
SW3         0013          -200
SYSRAM      4FCE          -639
TARRX       4FA5          -744  1042    1389
```

```
TARRY     4FA6         -745  1044  1391
TDOPT     0024         -680   828
TDOWN     16F4        -1270  1350
TDPACK    1788        -1364
THETBL    1459         -898   940
TICKIT    164A        -1144   933   949   965
TIMOUT    4FEC         -626
TLEFT     16CA        -1246
TMR60     4FEB         -625
TONEA     0011         -177
TONEB     0012         -178
TONEC     0013         -179   981
TONMO     0010         -176   985
TRIGHT    16D8        -1254  1334
TSTBIT    149C         -915   975
TUP       16E6        -1262  1342
UDMOVE    0003         -669  1215  1217
UMARGT    4FFB         -637
UNCRAM    4F96         -738   753   757   773
UPISTR    0000         -215
UPMUZK    1491         -911   918  1047
USERTB    4FFD         -638
VBBLNK    0006          -87
VBCCHK    0004          -84
VBCH      0003          -83
VBCL      0002          -82
VBCLAT    0003          -91
VBCLMT    0000          -89
VBCREV    0001          -90
VBDCH     0001          -81
VBDCL     0000          -80
VBDXH     0004          -68
VBDXL     0003          -67
VBDYH     0009          -73
VBDYL     0008          -72
VBLANK    0028         -242
VBMR      0000          -64
VBOAH     000E          -78
VBOAL     000D          -77
VBSACT    0007          -86
VBSTAT    0001          -65
VBTIMB    0002          -66
VBXCHK    0007          -71
VBXH      0006          -70
VBXL      0005          -69
VBYCHK    000C          -76
VBYH      000B          -75
VBYL      000A          -74
VECT      003E         -255
VECTC     003C         -254
VERAF     000E         -194
VERBL     000A         -174
VIBRA     0014         -180
VOICES    4FD4         -600
VOLAB     0016         -181
VOLC      0015         -182   983
VOLN      0017         -183
```

```
VWRITR    001E        -237
WASTE     0FFF        -585
WASTER    0FFF        -586
WIDTH     0004        -717  1338  1346
WPONOF    0000        -727
WPOPT     0001        -728
WPPAH     0003        -730
WPPAL     0002        -729
WPXSIZ    0005        -731
WPYSIZ    0004        -732
WRIT      0024        -240   883   883  1042  1042  1079  1079  1198
                      1198
WRITA     0026        -241
WRITOR    0010        -682  1040  1076  1196
WRITP     0022        -239
WRITR     0020        -238
XINTC     0002        -217   812
XMAX      009C        -673  1344
XPAND     0019        -191
XPNDON    0001         -35
XTAB1     0028        -697   698   699   813
XTAB2     0050        -698   817   819
XTAB3     0078        -699   815
YLINES    0015        -674   676   700   719   790
YMAX      005B        -676  1360
YTAB      0014        -700   701   702   703
YTAB1     001F        -701   817
YTAB2     0033        -702   813   815
YTAB3     0047        -703   819
ZSW       14C8        -943
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                641
                642            LIST S,M,X,T
                643            ORG  17DEH
17DE C3E819     644            JP   INIT


                646  ; *********************
                647  ; * GUN FIGHT EQUATES *
                648  ; *********************
                649   ; GUNFIGHT BACKGROUND JOB
                650   ; CONSISTING OF INITIALIZATION, PRE-ROUND DISPLAY,
                651   ; MONITORING OF CONTROLS AND VECTOR DELTA CHANGING
                652   ; DEATH, POST ROUND STUFF AND END GAME


                654   ;EQUATES
>0008           655  LNX      EQU  8              ; LEFT NUMBER X
>0002           656  BSY      EQU  2              ; BANNER STRINGS Y
>0088           657  RNX      EQU  136            ; RIGHT NUMBER X
>0020           658  LBULX    EQU  32             ; LEFT BULLETS X
>0068           659  RBULX    EQU  104            ; RIGHT " "
>004C           660  STMRX    EQU  76             ; SUB TIMER X
>002C           661  GRX      EQU  44             ; GET READY X
>0001           662  GRY      EQU  1              ; " Y
>0040           663  DRX      EQU  64             ; DRAW X
>0014           664  TCACY    EQU  20             ; TOP CACTUS Y
>000F           665  TTREEY   EQU  TCACY-5
>002A           666  MCACY    EQU  42             ; MID CACTUS Y
>0046           667  BCACY    EQU  70             ; BOTTOM CACTUS Y
>0041           668  BTREEY   EQU  BCACY-5
>0040           669  LCACX    EQU  64             ; LEFT CACTUS X
>0058           670  RCACX    EQU  88             ; RIGHT CACTUS X
>004C           671  CCACX    EQU  76             ; CENTER CACTUS X
>0048           672  WAGX     EQU  72             ; WAGON X
>0060           673  COWX     EQU  RCACX+8        ; OTHER COWBOYS WINDOW X
                674  ;
>000A           675  TLINE    EQU  10             ; TOP LINE OF GUNSPACE
>0009           676  ALINE    EQU  TLINE-1
>005C           677  BLINE    EQU  92             ; BOTTOM LINE OF "
                678  ;
>0012           679  BULVSZ   EQU  18             ; BULLET VECTOR SIZE
>0017           680  GFVSIZ   EQU  23
>0012           681  WAGVSZ   EQU  18             ; WAGON VECTOR SIZE
                682  ;
>0032           683  WINBND   EQU  50             ; TOP-BOTTOM WINDOW BOUNDARY
>006A           684  TOPLIN   EQU  53*2           ; TOP WINDOW LINE
>0000           685  BOTLIN   EQU  00             ; BOTTOM WINDOW LINE
>00C8           686  LFRLIN   EQU  100*2          ; LOW PRIORITY FOREGROUND LINE
                687  ;
>FFFF           688  NEXT     EQU  -1             ; NEXT LINK FOR QUEUES
>000F           689  VBARM    EQU  VBOAH+1        ; ARM STATE
>0010           690  VBOARM   EQU  VBARM+1        ; LAST ARM PATTERN WRITTEN
>0011           691  VBLEGT   EQU  VBOARM+1       ; LEG TIMER
>0012           692  VBLEG    EQU  VBLEGT+1       ; LEG LINK
>0013           693  VBCOMP   EQU  VBLEG+1        ; TIMER FOR COMPUTER CONTROL
```

```
                    694  ; BITS
>0000               695  VBSWAG   EQU   0              ; WAGON BIT
>0003               696  VBSCHG   EQU   3              ; CHANGE STATUS BIT
>0004               697  VBSNOM   EQU   4              ; NOT MOVING STATUS
>0005               698  VBSINT   EQU   5              ; INTERCEPTED/DEAD STATUS



                    700  ; ***************
                    701  ; * SUBROUTINES *
                    702  ; ***************
                    703  ; DISPLAY CLOCK AND UPDATE CT4
 17E1 F3            704  DCLOCK   DI
 17E2               705           SYSSUK DECCTS
 17E2 FF            705 +         RST   56
 17E3 11            705 +         DEFB  DECCTS+1
                    705 +         IF    DECCTS.EQ.INTPC
                    705 +         ENDIF
 17E4 80            706           DEFB  10000000B
 17E5 DD210D02      707           LD    IX,FNTSML
 17E9 3ADC4F        708           LD    A,(CT7)
 17EC B7            709           OR    A
 17ED 2808          710           JR    Z,DCOUT-$
 17EF               711           SYSSUK DISNUM
 17EF FF            711 +         RST   56
 17F0 37            711 +         DEFB  DISNUM+1
                    711 +         IF    DISNUM.EQ.INTPC
                    711 +         ENDIF
 17F1 4C            712           DEFB  STMRX
 17F2 02            713           DEFB  BSY
 17F3 0B            714           DEFB  TIME
 17F4 42            715           DEFB  42H
 17F5 DC4F          716           DEFW  CT7
 17F7 AF            717  DCOUT    XOR   A
 17F8 D30C          718           OUT   (MAGIC),A
 17FA 32FF0F        719           LD    (WASTE),A
 17FD FB            720           EI
 17FE C9            721           RET
                    722  ; FIRE BULLETS
                    723  ; LEFT COWBOY
 17FF               724  FIRE0    SYSSUK SUCK
 17FF FF            724 +         RST   56
 1800 0D            724 +         DEFB  SUCK+1
                    724 +         IF    SUCK.EQ.INTPC
                    724 +         ENDIF
 1801 DC            725           DEFB  11011100B
 1802 614F          726           DEFW  LCOWB
 1804 DA4F          727           DEFW  LBULS
 1806 194F          728           DEFW  BULV1+1
 1808 1809          729           JR    ZORE-$
 180A               730  FIRE1    SYSSUK SUCK
 180A FF            730 +         RST   56
 180B 0D            730 +         DEFB  SUCK+1
                    730 +         IF    SUCK.EQ.INTPC
                    730 +         ENDIF
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

180C DC         731           DEFB 11011100B
180D 784F       732           DEFW RCOWB
180F DB4F       733           DEFW RBULS
1811 3D4F       734           DEFW BULV3+1
1813 FD7E07     735  ZORE:    LD   A,(IY+CBB)
1816 B7         736           OR   A
1817 C8         737           RET  Z
1818 0A         738           LD   A,(BC)        ; GET BULLET COUNT
1819 B7         739           OR   A
181A C8         740           RET  Z
181B 7E         741           LD   A,(HL)        ; CHECK IF BULLET IS AVAILABLE
181C B7         742           OR   A
181D 2809       743           JR   Z,ZOK-$
181F 111200     744           LD   DE,BULVSZ     ; DELTA TO NEXT BULLET
1822 19         745           ADD  HL,DE
1823 7E         746           LD   A,(HL)
1824 B7         747           OR   A
1825 2801       748           JR   Z,ZOK-$
1827 C9         749           RET
                750  ; NOW HL->BULLET
                751  ;        IX->COWBOY
                752  ; SUB 1 FROM BULLET COUNT
1828 0A         753  ZOK      LD   A,(BC)
1829 3D         754           DEC  A
182A 02         755           LD   (BC),A
                756  ; SET SUB TIMER IF OUT OF BULLETS
182B 200D       757           JR   NZ,BERASE-$
182D 3ADC4F     758           LD   A,(CT7)
1830 B7         759           OR   A
1831 3E10       760           LD   A,10H
1833 2802       761           JR   Z,STSEC-$
1835 3E02       762           LD   A,2
1837 32DC4F     763  STSEC    LD   (CT7),A
183A E5         764  BERASE   PUSH HL
183B DDE5       765           PUSH IX
183D 0A         766           LD   A,(BC)
183E 6F         767           LD   L,A
183F 2600       768           LD   H,0
1841 29         769           ADD  HL,HL
1842 29         770           ADD  HL,HL         ;*4
1843 116802     771           LD   DE,BSY*256+RBULX
1846 DDCB0076   772           BIT  MRFLOP,(IX+VBMR)
184A 3E40       773           LD   A,40H         ; FLOPED MR
184C 2801       774           JR   Z,RITB-$
184E AF         775           XOR  A             ; NORMAL MR
                776  ; NOW POSITION AND ERASE
184F 19         777  RITB     ADD  HL,DE
1850 EB         778           EX   DE,HL
1851            779           SYSTEM RELAB1
1851 FF         779 +         RST  56
1852 3A         779 +         DEFB RELAB1
                779 +         IF   RELAB1.EQ.INTPC
                779 +         ENDIF
1853 EB         780           EX   DE,HL
1854 0605       781           LD   B,5
1856 112800     782           LD   DE,40         ; INC TO NEXT LINE
1859 36FF       783  BELP     LD   (HL),0FFH     ;ERASE A LINE
```

```
185B 19         784           ADD  HL,DE          ; GO DOWN A LINE
185C 10FB       785           DJNZ BELP-$
185E 1600       786           LD   D,0
1860 DD5E0F     787           LD   E,(IX+VBARM) ; GET CURRENT ARM POS
1863 62         788           LD   H,D
1864 6B         789           LD   L,E
1865 29         790           ADD  HL,HL          ;*2
1866 19         791           ADD  HL,DE          ; *3
1867 11931D     792           LD   DE,BULTAB
186A 19         793           ADD  HL,DE          ; -> BULTAB(ARM)
186B EB         794           EX   DE,HL
186C C1         795           POP  BC             ; BC<==IX
186D E1         796           POP  HL             ; BUL [STAT]
186E E5         797           PUSH HL             ; SAVE FOR ACTIVATE
186F 23         798           INC  HL             ; BUL [DEL TIME]
1870 3601       799           LD   (HL),1         ; MAKE BULIT JUMP OUT
1872 23         800           INC  HL             ; BUL [DEL XLOW]
1873 03         801           INC  BC             ; COW [STAT]
1874 03         802           INC  BC             ; COW [DEL TIME]
1875 03         803           INC  BC             ; COW [DX LO]
1876 CDD319     804           CALL PUTVEC
1879 03         805           INC  BC             ; COW [XCHK]
187A 03         806           INC  BC             ; COW [DY LO]
187B 23         807           INC  HL             ; BUL [XCHK]
187C 3601       808           LD   (HL),1         ; LIMIT CHECK
187E 23         809           INC  HL             ; BUL [DY LO]
187F CDD319     810           CALL PUTVEC
1882 E1         811           POP  HL             ; BUL [STAT]
1883 3680       812           LD   (HL),80H       ; ACTIVE
1885            813           SYSSUK BMUSIC
1885 FF         813 +         RST  56
1886 13         813 +         DEFB BMUSIC+1
                813 +         IF   BMUSIC.EQ.INTPC
                813 +         ENDIF
1887 124F       814           DEFW MSTACK
1889 01         815           DEFB 00000001B      ; JUST NOISE
188A DB1F       816           DEFW GUNSHOT
188C C9         817           RET
                818  ; TAKE A COFFEE BREAK
188D            819  NBRK:    DONT PIZBRK         ; SEE IF I CARE
188D 48         819 +         DEFB PIZBRK
188E            820           DO   MRET
188E 09         820 +         DEFB MRET+1
                821  ; CONVERT JOYSTICKS
188F DD21614F   822  JOY0     LD   IX,LCOWB
1893 1804       823           JR   PJOY-$
1895 DD21784F   824  JOY1     LD   IX,RCOWB
                825  ;    CONVERT   JOYSTICKS
1899 DD4E00     826  PJOY:    LD   C,(IX+VBMR)
189C 118000     827           LD   DE,128
189F 218000     828           LD   HL,128
18A2            829           SYSTEM MSKTD        ; COMPUTE DELTAS
18A2 FF         829 +         RST  56
18A3 7E         829 +         DEFB MSKTD
                829 +         IF   MSKTD.EQ.INTPC
                829 +         ENDIF
18A4 DD7409     830  STHN     LD   (IX+VBDYH),H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

18A7 DD7508     831           LD   (IX+VBDYL),L
18AA DD7204     832           LD   (IX+VBDXH),D
18AD DD7303     833           LD   (IX+VBDXL),E
18B0 C9         834           RET
18B1 DD21784F   835  PPOT1:   LD   IX,RCOWB
18B5 78         836           LD   A,B             ; POT MUST BE FLOPPED BECAUSE
18B6 2F         837           CPL                  ; ARM IS FLOPPED
18B7 1805       838           JR   PPOT-$
18B9 DD21614F   839  PPOTO:   LD   IX,LCOWB
18BD 78         840           LD   A,B
                841  ; CONVERT POT AND STORE
18BE E6E0       842  PPOT     AND  0E0H
18C0 0F         843           RRCA
18C1 0F         844           RRCA
18C2 0F         845           RRCA
18C3 0F         846           RRCA
18C4 FE0E       847           CP   0EH
18C6 2002       848           JR   NZ,KART-$
18C8 3E0C       849           LD   A,0CH           ; IF KNOB=7 THEN SET TO 6
18CA DD770F     850  KART     LD   (IX+VBARM),A ; SET ARM POSITION
18CD C9         851           RET
                852  ; CHECK IF BULLET HIT ANYTHING
18CE DD7E01     853  HITCHK:  LD   A,(IX+VBSTAT)
18D1 E660       854           AND  060H
18D3 FE20       855           CP   20H             ; CHECK ONLY IF BLANKED
18D5 280F       856           JR   Z,HIT-$
18D7 D0         857           RET  NC              ; RETURN IF NOT BLANKED YET
18D8 DDCB075E   858           BIT  VBCLAT,(IX+VBXCHK)
18DC C8         859           RET  Z
18DD DD360100   860           LD   (IX+VBSTAT),0 ;  BULLET HIT WALL
18E1 DD360701   861           LD   (IX+VBXCHK),1 ;  SET LIMIT CHECK
18E5 C9         862           RET
18E6 DD7E06     863  HIT:     LD   A,(IX+VBXH)   ;CHECK WHAT PART OF SCR ITS IN
18E9 FE48       864           CP   WAGX
18EB 300E       865           JR   NC,HIT1-$
18ED DD360202   866           LD   (IX+VBTIMB),2 ;MAKE IT JUMP OUT
18F1 DD360180   867           LD   (IX+VBSTAT),80H ;RE ACTIVATE
18F5 218F1D     868           LD   HL,BULLMT
18F8            869           SYSTEM VECT
18F8 FF         869 +         RST  56
18F9 3E         869 +         DEFB VECT
                869 +         IF   VECT.EQ.INTPC
                869 +         ENDIF
18FA C9         870           RET
18FB DD360100   871  HIT1:    LD   (IX+VBSTAT),0 ;  BULIT DIES FROM WAGON ON
18FF FE58       872           CP   RCACX
1901 301D       873           JR   NC,HIT2-$
1903 3A904F     874           LD   A,(WAGON)
1906 B7         875           OR   A               ; IS IT A CACTII?
1907 C0         876           RET  NZ              ; NOPE ITS A WAGON
1908 1E4C       877           LD   E,CCACX         ; LOAD X
                878  ; ERASE OBJECT BULLET HITS
190A DD560B     879  ERASE    LD   D,(IX+VBYH)   ; LOAD Y
190D 15         880           DEC  D
190E            881           SYSSUK RELAB1
190E FF         881 +         RST  56
190F 3B         881 +         DEFB RELAB1+1
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                881 +         IF   RELAB1.EQ.INTPC
                881 +         ENDIF
1910 00         882           DEFB 0
1911 EB         883           EX   DE,HL
1912 11D7FF     884           LD   DE,-41
1915 0600       885           LD   B,0
1917 7E         886  ELOP     LD   A,(HL)
1918 70         887           LD   (HL),B         ;ZERO THE SCREEN BYTE
1919 23         888           INC  HL
191A B6         889           OR   (HL)
191B 70         890           LD   (HL),B
191C 19         891           ADD  HL,DE
191D 20F8       892           JR   NZ,ELOP-$
191F C9         893           RET
1920 FE60       894  HIT2:    CP   RCACX+8        ; GUNFTR SPACE
1922 300C       895           JR   NC,DIE-$
1924 1E40       896           LD   E,LCACX
1926 DDCB0076   897           BIT  MRFLOP,(IX+VBMR)
192A 20DE       898           JR   NZ,ERASE-$
192C 1E58       899           LD   E,RCACX
192E 18DA       900           JR   ERASE-$
1930 DDCB0076   901  DIE:     BIT  MRFLOP,(IX+VBMR) ; WHO DIED?
1934 280C       902           JR   Z,DLEFT-$
1936            903           SYSSUK SUCK
1936 FF         903 +         RST  56
1937 0D         903 +         DEFB SUCK+1
                903 +         IF   SUCK.EQ.INTPC
                903 +         ENDIF
1938 DD         904           DEFB 11011101B
1939 614F       905           DEFW LCOWB
193B 08         906           DEFB 8
193C B11F       907           DEFW TAPS
193E A64F       908           DEFW RSCORE
1940 180A       909           JR   DIE1-$
1942            910  DLEFT    SYSSUK SUCK
1942 FF         910 +         RST  56
1943 0D         910 +         DEFB SUCK+1
                910 +         IF   SUCK.EQ.INTPC
                910 +         ENDIF
1944 DD         911           DEFB 11011101B
1945 784F       912           DEFW RCOWB
1947 64         913           DEFB 100
1948 C11F       914           DEFW FUNERL
194A A24F       915           DEFW LSCORE
194C DD361106   916  DIE1:    LD   (IX+VBLEGT),6 ;  SET FIRST CELL TIME
1950 DD361284   917           LD   (IX+VBLEG),KIL1.AND.0FFH ;??
1954 DD360168   918           LD   (IX+VBSTAT),068H ;  KILL HIM
1958 DD7E0B     919           LD   A,(IX+VBYH)   ;   WHERE TO WRITE GOT ME
195B D608       920           SUB  8
195D FE13       921           CP   TLINE+9
195F 3002       922           JR   NC,DIE4-$
1961 C620       923           ADD  A,32
1963 57         924  DIE4     LD   D,A            ; LOAD Y
1964            925           SYSTEM INCSCR
1964 FF         925 +         RST  56
1965 54         925 +         DEFB INCSCR
                925 +         IF   INCSCR.EQ.INTPC
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

                925 +         ENDIF
1966 2B         926           DEC  HL
1967 7E         927           LD   A,(HL)        ; FIELD
1968 FE05       928           CP   5             ; INC IF LESS THAN 5
196A CE00       929           ADC  A,0
196C 77         930           LD   (HL),A
                931  ; PLAY DEATH SONG
196D 60         932           LD   H,B
196E 69         933           LD   L,C
196F DD21124F   934           LD   IX,MSTACK
1973 3EC0       935           LD   A,11000000B
1975            936           SYSTEM BMUSIC
1975 FF         936 +         RST  56
1976 12         936 +         DEFB BMUSIC
                936 +         IF   BMUSIC.EQ.INTPC
                936 +         ENDIF
1977 0E0C       937           LD   C,LARG2
1979 21061F     938           LD   HL,GOTME
197C F3         939           DI
197D            940           SYSTEM STRDIS
197D FF         940 +         RST  56
197E 34         940 +         DEFB STRDIS
                940 +         IF   STRDIS.EQ.INTPC
                940 +         ENDIF
197F            941           SYSSUK PAWS
197F FF         941 +         RST  56
1980 51         941 +         DEFB PAWS+1
                941 +         IF   PAWS.EQ.INTPC
                941 +         ENDIF
1981 FA         942           DEFB 250
1982 3E01       943           LD   A,1
1984 32DE4F     944           LD   (SEMI4S),A    ; SET FLAGO
1987 C9         945           RET
                946  ; FIELD PUTS UP THE CACTII APPROP TO SCORE
                947  ; A=SCORE OF OPP PLAYER UPTO 6
                948  ; BC -> ARRAY OF Y POSITIONS
1988 21F81E     949  FIELD:   LD   HL,CACTUS     ; -> CACTUS PATTERN
198B F5         950           PUSH AF
198C 3E08       951           LD   A,1000B
198E D319       952           OUT  (XPAND),A
1990 F1         953           POP  AF
1991 FE01       954           CP   1
1993 D8         955           RET  C
1994 FE04       956           CP   4
1996 3003       957           JR   NC,TCAC-$
1998 CDC819     958           CALL CACW
199B 03         959  TCAC     INC  BC
199C FE02       960           CP   2
199E D8         961           RET  C
199F FE05       962           CP   5
19A1 3003       963           JR   NC,MCAC-$
19A3 CDC819     964           CALL CACW
19A6 FE03       965  MCAC     CP   3
19A8 D8         966           RET  C
19A9 03         967           INC  BC
19AA 08         968           EX   AF,AF'
19AB 3E81       969           LD   A,81H         ; ACTIVATE WAGON
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

19AD 32904F     970           LD   (WAGON),A
19B0 08         971           EX   AF,AF'
19B1 CDC819     972           CALL CACW
19B4 FE04       973           CP   4
19B6 D8         974           RET  C
19B7 03         975           INC  BC
19B8 21E91D     976           LD   HL,TREE
19BB F5         977           PUSH AF
19BC 3E0C       978           LD   A,1100B
19BE D319       979           OUT  (XPAND),A
19C0 F1         980           POP  AF
19C1 CDC819     981           CALL CACW
19C4 FE05       982           CP   5
19C6 D8         983           RET  C
19C7 03         984           INC  BC
19C8 F5         985  CACW:    PUSH AF
19C9 D5         986           PUSH DE
19CA 0A         987           LD   A,(BC)
19CB 57         988           LD   D,A
19CC 3E08       989           LD   A,8           ; EXPAND
19CE            990           SYSTEM WRITP
19CE FF         990 +         RST  56
19CF 22         990 +         DEFB WRITP
                990 +         IF   WRITP.EQ.INTPC
                990 +         ENDIF
19D0 D1         991           POP  DE
19D1 F1         992           POP  AF
19D2 C9         993           RET
                994  ; PUT DEL X,Y INTO BULLET VECTORS
19D3 1A         995  PUTVEC   LD   A,(DE)        ; TABLE [D LO]
19D4 77         996           LD   (HL),A        ; BUL [D LO]
19D5 13         997           INC  DE            ; TAB [D HI]
19D6 03         998           INC  BC            ; COW [D HI]
19D7 23         999           INC  HL            ; BUL [D HI]
19D8 1A        1000           LD   A,(DE)
19D9 77        1001           LD   (HL),A
19DA 23        1002           INC  HL            ; BUL [LO]
19DB 13        1003           INC  DE            ; TAB [HI]
19DC 03        1004           INC  BC            ; COW [LO]
19DD 3600      1005           LD   (HL),0
19DF 03        1006           INC  BC            ; COW [HI]
19E0 23        1007           INC  HL            ; BUL [HI]
19E1 0A        1008           LD   A,(BC)
19E2 EB        1009           EX   DE,HL
19E3 86        1010           ADD  A,(HL)
19E4 EB        1011           EX   DE,HL
19E5 77        1012           LD   (HL),A        ; BUL [HI]=COW [HI]+TAB [HI]
19E6 13        1013           INC  DE            ; TAB [D HI]
19E7 C9        1014           RET
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

               1016  ; GUNFIGHT START UP ROUTINE (ONCE PER GAME)
19E8           1017  INIT:    SYSSUK GETPAR
19E8 FF        1017 +         RST  56
19E9 4D        1017 +         DEFB GETPAR+1
               1017 +         IF   GETPAR.EQ.INTPC
               1017 +         ENDIF
19EA 1E02      1018           DEFW MXSCR
19EC 84        1019           DEFB 84H
19ED F44F      1020           DEFW ENDSCR
19EF 31064F    1021           LD   SP,STACK
19F2           1022           SYSTEM INTPC
19F2 FF        1022 +         RST  56
19F3 00        1022 +         DEFB INTPC
               1022 +         IF   INTPC.EQ.INTPC
>0001          1022 +INTP@    DEFL 1
               1022 +         ENDIF
19F4           1023           DO   FILL
19F4 1B        1023 +         DEFB FILL+1
19F5 064F      1024           DEFW STACK
19F7 D600      1025           DEFW CT7-STACK
19F9 00        1026           DEFB 0
19FA           1027           DO   SETB
19FA 7B        1027 +         DEFB SETB+1
19FB 02        1028           DEFB 2**GSBSCR
19FC F84F      1029           DEFW GAMSTB
19FE           1030           DO   SETOUT         ; SET UP GAME PORTS
19FE 17        1030 +         DEFB SETOUT+1
19FF B8        1031           DEFB BLINE*2        ; BOTTOM LINE - VERT BLK
1A00 D6        1032           DEFB RCACX/4+0C0H ; HORZ BOUNDS
1A01 08        1033           DEFB 8              ; INMOD
1A02           1034           DO   COLSET
1A02 19        1034 +         DEFB COLSET+1
1A03 C71D      1035           DEFW GFCOLS
1A05           1036           DO   BMUSIC         ; PLAY STREETS OF LOREDO
1A05 13        1036 +         DEFB BMUSIC+1
1A06 124F      1037           DEFW MSTACK
1A08 C0        1038           DEFB 11000000B      ; ON VOICE A
1A09 A31F      1039           DEFW HOME
1A0B           1040           EXIT
1A0B 02        1040 +         DEFB XINTC
>0000          1040 +INTP@    DEFL 0
               1041  ; *****
               1042  ; ONCE A ROUND START UP ROUTINE
               1043  ; *****
1A0C F3        1044  STRND:   DI
1A0D           1045           SYSTEM INTPC
1A0D FF        1045 +         RST  56
1A0E 00        1045 +         DEFB INTPC
               1045 +         IF   INTPC.EQ.INTPC
>0001          1045 +INTP@    DEFL 1
               1045 +         ENDIF
               1046  ; INIT HANDLES,BULLETS,TIMERS
1A0F           1047           DO   MOVE
1A0F 5F        1047 +         DEFB MOVE+1
1A10 DA4F      1048           DEFW CT5
1A12 0C00      1049           DEFW 12
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

1A14 CF1D      1050.          DEFW SINIT
               1051  ; COLOR BANNER
1A16           1052           FILL? NORMEM,BYTEPL*ALINE,OFFH
1A16 1B        1052 +         DEFB FILL+1
1A17 0040      1052 +         DEFW NORMEM
1A19 6801      1052 +         DEFW BYTEPL*ALINE
1A1B FF        1052 +         DEFB OFFH
               1053  ; ERASE SCREEN
1A1C           1054           FILL? NORMEM+BYTEPL*ALINE,BYTEPL*(BLINE-ALINE),0
1A1C 1B        1054 +         DEFB FILL+1
1A1D 6841      1054 +         DEFW NORMEM+BYTEPL*ALINE
1A1F F80C      1054 +         DEFW BYTEPL*(BLINE-ALINE)
1A21 00        1054 +         DEFB 0
               1055  ; RESET VECTORS
1A22           1056           FILL? STRRAM,ENDRAM-STRRAM,0
1A22 1B        1056 +         DEFB FILL+1
1A23 124F      1056 +         DEFW STRRAM     .
1A25 8F00      1056 +         DEFW ENDRAM-STRRAM
1A27 00        1056 +         DEFB 0
               1057  ; SHOW SCORES
1A28           1058           DO   SUCK
1A28 0D        1058 +         DEFB SUCK+1
1A29 10        1059           DEFB 00010000B     ; IX
1A2A 0D02      1060           DEFW FNTSML
1A2C           1061           DO   DISNUM
1A2C 37        1061 +         DEFB DISNUM+1
1A2D 08        1062           DEFB LNX
1A2E 02        1063           DEFB BSY
1A2F 0B        1064           DEFB TIME
1A30 C4        1065           DEFB 0C4H          ;ZERO SUPRS,SMALL
1A31 A24F      1066           DEFW LSCORE
1A33           1067           DO   DISNUM
1A33 37        1067 +         DEFB DISNUM+1
1A34 88        1068           DEFB RNX
1A35 02        1069           DEFB BSY
1A36 0B        1070           DEFB TIME
1A37 C4        1071           DEFB 0C4H
1A38 A64F      1072           DEFW RSCORE
               1073  ; CHECK FOR END GAME
1A3A           1074           DO   RCALL
1A3A 05        1074 +         DEFB RCALL+1
1A3B 301B      1075           DEFW ENDGAM
1A3D           1076           TEXT GETRDY,GRX,GRY,LARGE
1A3D 35        1076 +         DEFB STRDIS+1
1A3E 2C        1076 +         DEFB GRX
1A3F 01        1076 +         DEFB GRY
1A40 0B        1076 +         DEFB LARGE
1A41 7E1D      1076 +         DEFW GETRDY
1A43           1077           EXIT
1A43 02        1077 +         DEFB XINTC
>0000          1077 +INTP@    DEFL 0
1A44 AF        1078           XOR  A             ; SET UP WAGON
1A45 32904F    1079           LD   (WAGON),A     ; STOP WAGON
               1080  ; PUT UP PLAY FIELD:
1A48 3AA14F    1081           LD   A,(RFIELD)    ; NUMBER OF CACTII
1A4B 1E58      1082           LD   E,RCACX       ; RIGHT CAC COLUMN
1A4D 01C21D    1083           LD   BC,RFTAB      ; POSITIONS TABLE FOR CACTII
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

1A50 CD8819   1084           CALL FIELD          ; PUT THE CACTII UP
1A53 3AA54F   1085           LD   A,(LFIELD)
1A56 1E40     1086           LD   E,LCACX
1A58 01BD1D   1087           LD   BC,LFTAB
1A5B CD8819   1088           CALL FIELD
              1089  ; INITIALIZE Q POINTERS
1A5E 3E4F     1090  INITQ    LD   A,LCOWB.SHR.8
1A60 32144F   1091           LD   (WRITQ+2),A
1A63 32174F   1092           LD   (VECQ+2),A
              1093  ; SET UP VECTORS SO COWBOYS WALK OUT
1A66 DD21614F 1094           LD   IX,LCOWB       ; LEFT COMBOY VECTOR
1A6A DD360010 1095           LD   (IX+VBMR),10H
1A6E 21154F   1096           LD   HL,VECQ
1A71 CD341D   1097           CALL COWINT
1A74 DD21784F 1098           LD   IX,RCOWB       ; RIGHT COWBOY VECTOR
1A78 DD360050 1099           LD   (IX+VBMR),50H
1A7C CD341D   1100           CALL COWINT
1A7F 3A904F   1101           LD   A,(WAGON)      ; IF WAGON IS ON
1A82 B7       1102           OR   A
1A83 281D     1103           JR   Z,MIDC-$
1A85 DD218F4F 1104           LD   IX,WAGVEC      ; THEN ACTIVATE WAGON
1A89 DD360010 1105           LD   (IX+VBMR),10H
1A8D DD360C03 1106           LD   (IX+VBYCHK),3
1A91 DD360840 1107           LD   (IX+VBDYL),40H
1A95 DD360648 1108           LD   (IX+VBXH),72
1A99 DD360B0A 1109           LD   (IX+VBYH),TLINE
1A9D CD541D   1110           CALL ADDTQ
1AA0 180B     1111           JR   BORG-$
1AA2 3E08     1112  MIDC:    LD   A,8
1AA4 D319     1113           OUT  (XPAND),A
1AA6          1114           SYSSUK WRITP        ; ELSE PUT UP A CACTUS
1AA6 FF       1114 +         RST  56
1AA7 23       1114 +         DEFB WRITP+1
              1114 +         IF   WRITP.EQ.INTPC
              1114 +         ENDIF
1AA8 4C       1115           DEFB CCACX
1AA9 2A       1116           DEFB MCACY
1AAA 08       1117           DEFB 8              ; EXPAND
1AAB F81E     1118           DEFW CACTUS
              1119  ; INITIALIZE BULLET VECTORS
1AAD 111200   1120  BORG:    LD   DE,BULVSZ
1AB0 DD21184F 1121           LD   IX,BULV1
1AB4 012004   1122           LD   BC,4*256+20H
1AB7 3E02     1123           LD   A,2
1AB9 B8       1124  BULLP    CP   B
1ABA 2002     1125           JR   NZ,TIYU-$
1ABC 0E60     1126           LD   C,60H
1ABE DD7100   1127  TIYU     LD   (IX+VBMR),C
1AC1 DD360701 1128           LD   (IX+VBXCHK),1
1AC5 DD360C03 1129           LD   (IX+VBYCHK),3
1AC9 DD19     1130           ADD  IX,DE
1ACB 10EC     1131           DJNZ BULLP-$
              1132  ; FIRE UP INTERRUPTS
1ACD 3E1D     1133           LD   A,INTTBL.SHR.8
1ACF ED47     1134           LD   I,A
              1135  ;        IM     2          ;DONE IN MENU
1AD1 3E78     1136           LD   A,LFRVEC.AND.0FFH
```

```
 ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

 1AD3 D30D      1137           OUT  (INFBK),A
                1138  ; ***
                1139  ; LET COWBOYS WALK OUT
                1140  ; ***
 1AD5           1141  WALK:    SYSSUK PAWS
 1AD5 FF        1141 +         RST  56
 1AD6 51        1141 +         DEFB PAWS+1
                1141 +         IF   PAWS.EQ.INTPC
                1141 +         ENDIF
 1AD7 64        1142           DEFB 100
 1AD8 F3        1143           DI
 1AD9 DD210D02  1144           LD   IX,FNTSML
 1ADD           1145           SYSTEM INTPC
 1ADD FF        1145 +         RST  56
 1ADE 00        1145 +         DEFB INTPC
                1145 +         IF   INTPC.EQ.INTPC
>0001           1145 +INTP@    DEFL 1
                1145 +         ENDIF
                1146  ; ERASE GET READY
 1ADF           1147           DO   BLANK
 1ADF 2B        1147 +         DEFB BLANK+1
 1AE0 12        1148           DEFB 18
 1AE1 08        1149           DEFB 8
 1AE2 FF        1150           DEFB 0FFH
 1AE3 00000000  1151           XYDEFW (GRX/4)+4000H,GRY
 1AE7           1152           TEXT DRAW,DRX,GRY,LARGE
 1AE7 35        1152 +         DEFB STRDIS+1
 1AE8 40        1152 +         DEFB DRX
 1AE9 01        1152 +         DEFB GRY
 1AEA 0B        1152 +         DEFB LARGE
 1AEB 8B1D      1152 +         DEFW DRAW
 1AED           1153           DO   CHRDIS
 1AED 33        1153 +         DEFB CHRDIS+1
 1AEE 20        1154           DEFB LBULX
 1AEF 02        1155           DEFB BSY
 1AF0 0B        1156           DEFB BULT
 1AF1 BB        1157           DEFB 0BBH          ; BULLET
 1AF2           1158           DO   MCALL         ; 5 MORE
 1AF2 07        1158 +         DEFB MCALL+1
 1AF3 571B      1159           DEFW BULRIT
 1AF5           1160           DO   SUCK
 1AF5 0D        1160 +         DEFB SUCK+1
 1AF6 01        1161           DEFB 00000001B
 1AF7 68        1162           DEFB RBULX         ; DO THE RIGHT ONES
 1AF8           1163           DONT CHRDIS        ; DISPLAY FIRST ONE
 1AF8 32        1163 +         DEFB CHRDIS
 1AF9           1164           DO   MCALL         ; DISP THE OTHER 5
 1AF9 07        1164 +         DEFB MCALL+1
 1AFA 571B      1165           DEFW BULRIT
 1AFC           1166           DO   PAWS
 1AFC 51        1166 +         DEFB PAWS+1
 1AFD 3C        1167           DEFB 60
 1AFE           1168           DO   BLANK
 1AFE 2B        1168 +         DEFB BLANK+1
 1AFF 08        1169           DEFB 8
 1B00 08        1170           DEFB 8
 1B01 FF        1171           DEFB 0FFH
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND       COMMENT

1B02 00000000 1172           XYDEFW (DRX/4)+4000H,GRY
1B06          1173           EXIT
1B06 02       1173 +         DEFB XINTC
>0000         1173 +INTP@    DEFL 0
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               1175  ; *****
               1176  ; MAIN LOOP DURING ROUND
               1177  ; GETS HANDLES, SETS VECTORS AND CHECKS BULLETS
1B07           1178  LOOP:    SYSTEM INTPC
1B07 FF        1178 +         RST  56
1B08 00        1178 +         DEFB INTPC
               1178 +         IF   INTPC.EQ.INTPC
>0001          1178 +INTP@    DEFL 1
               1178 +         ENDIF
1B09           1179           DO   SENTRY
1B09 43        1179 +         DEFB SENTRY+1
1B0A 1402      1180           DEFW ALKEYS
1B0C           1181           DO   DOIT
1B0C 45        1181 +         DEFB DOIT+1
1B0D 381B      1182           DEFW DTAB
1B0F           1183           EXIT
1B0F 02        1183 +         DEFB XINTC
>0000          1183 +INTP@    DEFL 0


               1185  ; CHECK FOR DEATHS
1B10 DD21184F  1186  DEATH    LD   IX,BULV1
1B14 111200    1187           LD   DE,BULVSZ
1B17 0604      1188           LD   B,4
1B19 C5        1189  LPPP2    PUSH BC
1B1A D5        1190           PUSH DE
1B1B CDCE18    1191           CALL HITCHK
1B1E D1        1192           POP  DE
1B1F C1        1193           POP  BC
1B20 DD19      1194           ADD  IX,DE
1B22 3ADE4F    1195           LD   A,(SEMI4S)    ; CHECK IF DEATH MODE
1B25 3D        1196           DEC  A
1B26 28DF      1197           JR   Z,LOOP-$
1B28 10EF      1198           DJNZ LPPP2-$
1B2A 18DB      1199           JR   LOOP-$
               1200  ;
1B2C           1201  ENDRND   EXIT
1B2C 02        1201 +         DEFB XINTC
>0000          1201 +INTP@    DEFL 0
1B2D C30C1A    1202           JP   STRND
               1203  ;
1B30 3AF84F    1204  ENDGAM:  LD   A,(GAMSTB)
1B33 CB7F      1205           BIT  GSBEND,A
1B35 C8        1206           RET  Z
1B36           1207           SYSTEM QUIT
1B36 FF        1207 +         RST  56
1B37 78        1207 +         DEFB QUIT
               1207 +         IF   QUIT.EQ.INTPC
               1207 +         ENDIF



1B38           1209  DTAB:    JMP  SCT7,ENDRND
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND      COMMENT

1B38 08       1209 +        DEFB SCT7
1B39 2C1B     1209 +        DEFW ENDRND
              1209 +        IF   0
              1209 +        ENDIF
1B3B          1210          JMP  SF0,ENDRND
1B3B 09       1210 +        DEFB SF0
1B3C 2C1B     1210 +        DEFW ENDRND
              1210 +        IF   0
              1210 +        ENDIF
1B3E          1211          RC   SP0,PPOT0
1B3E 5C       1211 +        DEFB SP0+40H
1B3F B918     1211 +        DEFW PPOT0
              1211 +        IF   0
              1211 +        ENDIF
1B41          1212          RC   SP1,PPOT1
1B41 5D       1212 +        DEFB SP1+40H
1B42 B118     1212 +        DEFW PPOT1
              1212 +        IF   0
              1212 +        ENDIF
1B44          1213          RC   SJ0,JOY0
1B44 55       1213 +        DEFB SJ0+40H
1B45 8F18     1213 +        DEFW JOY0
              1213 +        IF   0
              1213 +        ENDIF
1B47          1214          RC   SJ1,JOY1
1B47 57       1214 +        DEFB SJ1+40H
1B48 9518     1214 +        DEFW JOY1
              1214 +        IF   0
              1214 +        ENDIF
1B4A          1215          MC   SKYD,NBRK
1B4A 93       1215 +        DEFB SKYD+80H
1B4B 8D18     1215 +        DEFW NBRK
              1215 +        IF   0
              1215 +        ENDIF
1B4D          1216          RC   ST0,FIRE0
1B4D 54       1216 +        DEFB ST0+40H
1B4E FF17     1216 +        DEFW FIRE0
              1216 +        IF   0
              1216 +        ENDIF
1B50          1217          RC   ST1,FIRE1
1B50 56       1217 +        DEFB ST1+40H
1B51 0A18     1217 +        DEFW FIRE1
              1217 +        IF   0
              1217 +        ENDIF
1B53          1218          RC   SSEC,DCLOCK,+END
1B53 51       1218 +        DEFB SSEC+40H
1B54 E117     1218 +        DEFW DCLOCK
              1218 +        IF   0+END
1B56 C0       1218 +        DEFB 0+END
              1218 +        ENDIF



1B57          1220  BULRIT  DONT CHRDIS
1B57 32       1220 +        DEFB CHRDIS
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND      COMMENT

1B58          1221          DONT CHRDIS
1B58 32       1221 +        DEFB CHRDIS
1B59          1222          DONT CHRDIS
1B59 32       1222 +        DEFB CHRDIS
1B5A          1223          DONT CHRDIS
1B5A 32       1223 +        DEFB CHRDIS
1B5B          1224          DONT CHRDIS
1B5B 32       1224 +        DEFB CHRDIS
1B5C          1225          DONT MRET
1B5C 08       1225 +        DEFB MRET



              1227  ; ************************************
              1228  ; * GUNFIGHT WRITE INTERRUPT ROUTINE *
              1229  ; ************************************
1B5D 08       1230  GFWRIT: EX   AF,AF'
1B5E D9       1231          EXX
1B5F DDE5     1232          PUSH IX
1B61 3E78     1233  BEGINT: LD   A,LFRVEC.AND.0FFH ;  ESTABLISH TICKS INT
1B63 D30D     1234          OUT  (INFBK),A
1B65 3EC8     1235          LD   A,LFRLIN
1B67 D30F     1236          OUT  (INLIN),A
1B69 21124F   1237          LD   HL,WRITQ     ; GET FIRST WRITE Q ENTRY
1B6C CD6B1D   1238          CALL FIRST
1B6F CD291D   1239          CALL DELQ         ; DROP FROM WRITE Q
1B72 AF       1240          XOR  A
1B73 32FF0F   1241          LD   (WASTE),A
1B76 DDCB0146 1242          BIT  VBSWAG,(IX+VBSTAT) ; WAGON?
1B7A 2028     1243          JR   NZ,GFWRT1-$  ; JUMP IF YEP
              1244  ; GUNFIGHTER - BLANKETH HIM
1B7C 110514   1245          LD   DE,1405H     ; LOAD BLANKING PARMS
1B7F          1246          SYSTEM VBLANK     ; CALL BLANKER
1B7F FF       1246 +        RST  56
1B80 28       1246 +        DEFB VBLANK
              1246 +        IF   VBLANK.EQ.INTPC
              1246 +        ENDIF
1B81 261E     1247          LD   H,LEGO.SHR.8 ; WRITE LEG PATTERN
1B83 DD6E12   1248          LD   L,(IX+VBLEG)
1B86 2C       1249          INC  L            ; SKIP OVER LINK AND TIME
1B87 2C       1250          INC  L
1B88          1251          SYSTEM VWRITR     ; AND WRITE LEG
1B88 FF       1251 +        RST  56
1B89 1E       1251 +        DEFB VWRITR
              1251 +        IF   VWRITR.EQ.INTPC
              1251 +        ENDIF
              1252  ; IS GUNFIGHTER DEAD?
1B8A DDCB016E 1253          BIT  VBSINT,(IX+VBSTAT)
1B8E 2030     1254          JR   NZ,GFWRT5-$  ; JUMP IF SO
1B90 21DB1D   1255          LD   HL,ARMTBL    ; LOOKUP ARM PATTERN
1B93 1600     1256          LD   D,0
1B95 DD5E0F   1257          LD   E,(IX+VBARM)
1B98 19       1258          ADD  HL,DE
1B99 5E       1259          LD   E,(HL)
1B9A 23       1260          INC  HL
```

```
ADDR OBJECT   STMT   LABEL   OPCD OPERAND       COMMENT

1B9B 56        1261           LD    D,(HL)
1B9C EB        1262           EX    DE,HL
1B9D           1263           SYSTEM VWRITR      ; WRITE ARM PATTERN
1B9D FF        1263 +         RST   56
1B9E 1E        1263 +         DEFB  VWRITR
               1263 +         IF    VWRITR.EQ.INTPC
               1263 +         ENDIF
1B9F 21101F    1264           LD    HL,GFBODY    ; LOAD BODY PATTERN
1BA2 1808      1265           JR    GFWRT2-$     ; JOIN WAGON WRITE
               1266   ; BLANK THE WAGON
1BA4 110416    1267   GFWRT1: LD    DE,1604H     ; LOAD WAGON SIZE
1BA7           1268           SYSTEM VBLANK
1BA7 FF        1268 +         RST   56
1BA8 28        1268 +         DEFB  VBLANK
               1268 +         IF    VBLANK.EQ.INTPC
               1268 +         ENDIF
1BA9 21401F    1269           LD    HL,WAGPAT
1BAC           1270   GFWRT2: SYSTEM VWRITR      ; NOW WRITE
1BAC FF        1270 +         RST   56
1BAD 1E        1270 +         DEFB  VWRITR
               1270 +         IF    VWRITR.EQ.INTPC
               1270 +         ENDIF
1BAE DD720E    1271   GFWRT4: LD    (IX+VBOAH),D
1BB1 DD730D    1272           LD    (IX+VBOAL),E
1BB4 21154F    1273   GFWRT3: LD    HL,VECQ      ; ADD VECTOR TO VECTOR Q
1BB7 CD541D    1274           CALL  ADDTQ
1BBA DDE1      1275           POP   IX
1BBC 08        1276           EX    AF,AF'
1BBD D9        1277           EXX
1BBE FB        1278   EIRE    EI
1BBF C9        1279           RET
1BC0 210C1F    1280   GFWRT5: LD    HL,NULPAT
1BC3 18E7      1281           JR    GFWRT2-$
               1282   ; ***********************************
               1283   ; * GUNFIGHT LOW FOREGROUND ROUTINE *
               1284   ; ***********************************
1BC5 F5        1285   GFLFR:  PUSH  AF
1BC6 C5        1286           PUSH  BC
1BC7 D5        1287           PUSH  DE
1BC8 E5        1288           PUSH  HL
1BC9 DDE5      1289           PUSH  IX
               1290   ; BUMP TIME BASES OF ACTIVE OR INTERCEPTED VECTORS
1BCB 21194F    1291           LD    HL,BULV1+VBSTAT
1BCE 111100    1292           LD    DE,BULVSZ-1
1BD1 0604      1293           LD    B,4
1BD3 CD1E1D    1294           CALL  TBUMP
1BD6 23        1295           INC   HL           ; SKIP LINK FIELD
1BD7 111600    1296           LD    DE,GFVSIZ-1
1BDA 0603      1297           LD    B,3
1BDC CD1E1D    1298           CALL  TBUMP
               1299   ; LOOP TO UNWRITE, THEN WRITE ALL 4 BULLETS
               1300   ; BUT FIRST, A WORD TO OUR SHIFTER
1BDF AF        1301           XOR   A
1BE0 32FF0F    1302           LD    (WASTE),A
1BE3 0604      1303           LD    B,4
1BE5 DD21184F  1304           LD    IX,BULV1
               1305   ; UNWRITE THIS GUY?
```

```
ADDR OBJECT   STMT  LABEL    OPCD OPERAND        COMMENT

1BE9 DDCB0176 1306  WRBUL1:  BIT  VBBLNK,(IX+VBSTAT)
1BED 2811     1307           JR   Z,WRBUL2-$    ; JUMP IF NOT
1BEF DD660E   1308           LD   H,(IX+VBOAH)
1BF2 DD6E0D   1309           LD   L,(IX+VBOAL)
1BF5 DD7E0F   1310           LD   A,(IX+VBARM) ; GET LAST MR
1BF8 D30C     1311           OUT  (MAGIC),A
1BFA 36C0     1312           LD   (HL),0C0H     ; UNWRITE BULLET
1BFC DDCB01B6 1313           RES  VBBLNK,(IX+VBSTAT) ; CLEAR BLANK BIT
              1314  ; SHALL WE WRITE THIS GUY?
1C00 DDCB017E 1315  WRBUL2:  BIT  VBSACT,(IX+VBSTAT)
1C04 282B     1316           JR   Z,WRBUL4-$
1C06 DD560B   1317           LD   D,(IX+VBYH)
1C09 DD5E06   1318           LD   E,(IX+VBXH)
1C0C DD7E00   1319           LD   A,(IX+VBMR)
1C0F          1320           SYSTEM RELABS
1C0F FF       1320 +         RST  56
1C10 38       1320 +         DEFB RELABS
              1320 +         IF   RELABS.EQ.INTPC
              1320 +         ENDIF
1C11 DD720E   1321           LD   (IX+VBOAH),D
1C14 DD730D   1322           LD   (IX+VBOAL),E
1C17 DD770F   1323           LD   (IX+VBARM),A
1C1A 210040   1324           LD   HL,NORMEM-SCREEN
1C1D 19       1325           ADD  HL,DE
>4FFF         1326  DIFER    EQU  WASTE-SCREEN+NORMEM
1C1E 7E       1327           LD   A,(HL)
1C1F EB       1328           EX   DE,HL
1C20 36C0     1329           LD   (HL),0C0H
1C22 B7       1330           OR   A
1C23 2808     1331           JR   Z,WRBUL3-$    ; JUMP IF NOT
1C25 DDCB01BE 1332           RES  VBSACT,(IX+VBSTAT) ; KILL ACTIVE BIT
1C29 DDCB01EE 1333           SET  VBSINT,(IX+VBSTAT) ; SET INTERCEPT BIT
1C2D DDCB01F6 1334  WRBUL3:  SET  VBBLNK,(IX+VBSTAT) ; SET BLANK BIT
              1335  ; STEP TO NEXT BULLET VECTOR, LOOP BACK IF NOT DONE
1C31 111200   1336  WRBUL4:  LD   DE,BULVSZ
1C34 DD19     1337           ADD  IX,DE
1C36 10B1     1338           DJNZ WRBUL1-$
              1339  ; GET NEXT PATTERN TO WRITE, AND SCHEDULE HIM
1C38 21124F   1340           LD   HL,WRITQ
1C3B CD6B1D   1341           CALL FIRST
1C3E 2812     1342           JR   Z,WRBL5A-$    ; JUMP IF EMPTY Q
1C40 3E7A     1343           LD   A,WRTVEC.AND.0FFH ; SET FEEDBACK REG
1C42 D30D     1344           OUT  (INFBK),A
1C44 DD7E0B   1345           LD   A,(IX+VBYH)   ; WHICH WINDOW TO USE?
1C47 FE32     1346           CP   WINBND        ; COMPARE TO WINDOW BOUNDARY
1C49 3E00     1347           LD   A,BOTLIN      ; ASSUME BOTTOM LINE
1C4B 3002     1348           JR   NC,WRBUL5-$   ; JUMP IF GOOD GUESS
1C4D 3E6A     1349           LD   A,TOPLIN      ; WRONG - USE TOP
1C4F D30F     1350  WRBUL5:  OUT  (INLIN),A     ; SET LINE REGISTER
1C51 FB       1351           EI
              1352  ; LOOP THRU VECTORING THOSE BULLETS
1C52 DD21184F 1353  WRBL5A   LD   IX,BULV1
1C56 0604     1354           LD   B,4
1C58 218F1D   1355           LD   HL,BULLMT     ; HL = BULLET LIMITS TABLE
1C5B 111200   1356           LD   DE,BULVSZ
1C5E DDCB017E 1357  WRBUL6:  BIT  VBSACT,(IX+VBSTAT) ; ACTIVE BULLET?
1C62 280C     1358           JR   Z,WRBUL7-$
```

```
ADDR OBJECT   STMT LABEL    OPCD OPERAND       COMMENT

1C64          1359          SYSTEM VECT
1C64 FF       1359 +        RST   56
1C65 3E       1359 +        DEFB  VECT
              1359 +        IF    VECT.EQ.INTPC
              1359 +        ENDIF
1C66 DDCB075E 1360          BIT   VBCLAT,(IX+VBXCHK) ; DID Y HIT EDGE?
1C6A 2804     1361          JR    Z,WRBUL7-$     ; NOPE
1C6C DDCB01BE 1362          RES   VBSACT,(IX+VBSTAT) ; DEACTIVATE BULLET
1C70 DD19     1363 WRBUL7:  ADD   IX,DE
1C72 10EA     1364          DJNZ  WRBUL6-$       ; LOOP BACK
              1365 ; NOW PUT SOMETHING ON THE WRITE Q
1C74 0602     1366          LD    B,2            ; MAX 2 TIMES THRU
1C76 21154F   1367          LD    HL,VECQ
1C79 CD6B1D   1368 GVECT:   CALL  FIRST          ; GET VECTOR Q ENTRY
1C7C CAFC1C   1369          JP    Z,GVECT4       ; JUMP IF Q EMPTY
1C7F CD291D   1370          CALL  DELQ           ; DROP FROM VECTOR Q
1C82 FB       1371          EI
              1372 ; WAGON?
1C83 DDCB0146 1373          BIT   VBSWAG,(IX+VBSTAT)
1C87 C2071D   1374          JP    NZ,GVECT5      ; JUMP ON WAGON
              1375 ; DEAD?
1C8A DDCB016E 1376          BIT   VBSINT,(IX+VBSTAT)
1C8E 2025     1377          JR    NZ,GVECT1-$    ; JUMP IF DEAD
              1378 ; ZERO VELOCITY?
1C90 DD7E03   1379          LD    A,(IX+VBDXL)
1C93 DDB604   1380          OR    (IX+VBDXH)
1C96 DDB608   1381          OR    (IX+VBDYL)
1C99 DDB609   1382          OR    (IX+VBDYH)
1C9C 2017     1383          JR    NZ,GVECT1-$    ; GVECT1 IF NONZERO
1C9E DD7702   1384          LD    (IX+VBTIMB),A ; ZERO TIME BASE
1CA1 DDCB0166 1385          BIT   VBSNOM,(IX+VBSTAT) ; ALREADY STATIONARY?
1CA5 2036     1386          JR    NZ,GVEC3A-$
              1387 ; SET STATIONARY LEGS
1CA7 DD36124F 1388          LD    (IX+VBLEG),LEGO.AND.OFFH
1CAB DDCB01DE 1389          SET   VBSCHG,(IX+VBSTAT) ; SET CHANGED
1CAF DDCB01E6 1390          SET   VBSNOM,(IX+VBSTAT) ;AND STATIONARY
1CB3 1828     1391          JR    GVEC3A-$       ; JUMP TO ARM CHECK
              1392 ; MOVING GUNFIGHTER
              1393 ; VECTOR
1CB5 21871D   1394 GVECT1:  LD    HL,GUNLMT      ; LOAD GF LIMITS
1CB8          1395          SYSTEM VECT
1CB8 FF       1395 +        RST   56
1CB9 3E       1395 +        DEFB  VECT
              1395 +        IF    VECT.EQ.INTPC
              1395 +        ENDIF
1CBA 2808     1396          JR    Z,GVECT2-$     ; JUMP IF HE DIDN'T MOVE
1CBC DDCB01DE 1397          SET   VBSCHG,(IX+VBSTAT) ; SET CHANGED BIT
1CC0 DDCB01A6 1398          RES   VBSNOM,(IX+VBSTAT) ; CLEAR NOT MOVING STATUS
              1399 ; NEED WE GO TO NEXT CELL IN ANIMATION SEQUENCE?
1CC4 DD7E11   1400 GVECT2:  LD    A,(IX+VBLEGT) ; A = ANIMATION TIMER
1CC7 91       1401          SUB   C              ; SUBTRACT TIME BASE
1CC8 F2DA1C   1402          JP    P,GVECT3       ; JUMP IF NOT COUNTED DOWN
              1403 ; GET NEXT CELL
1CCB DD5E12   1404          LD    E,(IX+VBLEG) ; GET LINK
1CCE 161E     1405          LD    D,LEGO.SHR.8 ; SET H.O. PART
1CD0 1A       1406          LD    A,(DE)         ; A = NEXT
1CD1 DD7712   1407          LD    (IX+VBLEG),A
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1CD4 13        1408           INC  DE              ; STEP TO TIMER
1CD5 1A        1409           LD   A,(DE)          ; GET NEW TIMER
1CD6 DDCB01DE  1410           SET  VBSCHG,(IX+VBSTAT) ; SET CHANGED BIT
1CDA DD7711    1411  GVECT3:  LD   (IX+VBLEGT),A ; STORE BACK TIMER
               1412  ; DID ARM CHANGE?
1CDD DD7E0F    1413  GVEC3A:  LD   A,(IX+VBARM)
1CE0 DDBE10    1414           CP   (IX+VBOARM)   ; COMPARE TO OLD ARM
1CE3 2807      1415           JR   Z,GVEC3B-$    ; JUMP IF NO CHANGE
1CE5 DDCB01DE  1416           SET  VBSCHG,(IX+VBSTAT) ; SET CHANGED BIT
1CE9 DD7710    1417           LD   (IX+VBOARM),A
               1418  ; ADD ITEM TO WRITE Q?
1CEC DDCB015E  1419  GVEC3B:  BIT  VBSCHG,(IX+VBSTAT)
1CF0 2020      1420           JR   NZ,GVECT6-$   ; YES GVECT6
               1421  ; NO CHANGE - LINK TO VECTOR Q
1CF2 21154F    1422           LD   HL,VECQ
1CF5 CD541D    1423           CALL ADDTQ
1CF8 05        1424           DEC  B
1CF9 C2791C    1425           JP   NZ,GVECT       ; SUB FOR DJNZ
1CFC FB        1426  GVECT4:  EI
1CFD CD0002    1427           CALL STIMER
1D00 DDE1      1428           POP  IX
1D02 E1        1429           POP  HL
1D03 D1        1430           POP  DE
1D04 C1        1431           POP  BC
1D05 F1        1432           POP  AF
1D06 C9        1433           RET
               1434  ; VECTOR AND Q WAGON
1D07 217C1D    1435  GVECT5:  LD   HL,WAGLMT
1D0A           1436           SYSTEM VECT
1D0A FF        1436 +         RST  56
1D0B 3E        1436 +         DEFB VECT
               1436 +         IF   VECT.EQ.INTPC
               1436 +         ENDIF
1D0C 21154F    1437           LD   HL,VECQ
1D0F CD291D    1438           CALL DELQ           ; REMOVE FROM VECTOR Q
1D12 DDCB019E  1439  GVECT6:  RES  VBSCHG,(IX+VBSTAT)
1D16 21124F    1440           LD   HL,WRITQ
1D19 CD541D    1441           CALL ADDTQ
1D1C 18DE      1442           JR   GVECT4-$       ; JUMP BACK TO QUIT
               1443  ; ROUTINE TO BUMP TIME BASES OF VECTORS
1D1E 7E        1444  TBUMP:   LD   A,(HL)         ; GET STATUS
1D1F 23        1445           INC  HL
1D20 E6A0      1446           AND  0A0H           ; ACTIVE OR INTERCEPTED?
1D22 2801      1447           JR   Z,TBUMP1-$     ; NO - TBUMP1
1D24 34        1448           INC  (HL)           ; BUMP THE TIME BASE
1D25 19        1449  TBUMP1:  ADD  HL,DE
1D26 10F6      1450           DJNZ TBUMP-$
1D28 C9        1451           RET
               1452  ; SUBROUTINE TO DELETE ENTRY AT FRONT OF Q
               1453  ; ENTRY:          HL = HEAD-TAIL,  IX = OBJECT,  A = CLOBBERE
1D29 F3        1454  DELQ:    DI
1D2A DD7EFF    1455           LD   A,(IX+NEXT)   ; HEAD = NEXT(OBJECT)
1D2D 77        1456           LD   (HL),A
1D2E A7        1457           AND  A              ; IS HEAD NOW NIL?
1D2F C0        1458           RET  NZ             ; QUIT IF NOT
1D30 23        1459           INC  HL             ; YES - SET TAIL = NIL TOO
1D31 77        1460           LD   (HL),A
```

```
1D32 2B        1461           DEC  HL
1D33 C9        1462           RET
1D34 DD360332  1463  COWINT   LD   (IX+VBDXL),50 ;  SLOW WALK OUT
1D38 DD360180  1464           LD   (IX+VBSTAT),80H ;  ACTIVATE
1D3C DD360701  1465           LD   (IX+VBXCHK),1
1D40 DD360C01  1466           LD   (IX+VBYCHK),1
1D44 DD360604  1467           LD   (IX+VBXH),4
1D48 DD360B28  1468           LD   (IX+VBYH),40
1D4C DD360F06  1469           LD   (IX+VBARM),6 ;  SET ARM STRAIGHT
1D50 DD36124F  1470           LD   (IX+VBLEG),LEGO.AND.OFFH
               1471  ;        JP      ADDTQ
               1472  ;  SUBROUTINE TO APPEND ENTRY TO END OF Q
               1473  ;  ENTRY:         HL = HEAD-TAIL BYTES, IX = OBJECT, A,DE C
1D54 DDE5      1474  ADDTQ:   PUSH IX            ;  DE = ENTRY
1D56 D1        1475           POP  DE
1D57 F3        1476           DI
1D58 DD36FF00  1477           LD   (IX+NEXT),0   ;  NEXT(OBJ)=NIL
1D5C 23        1478           INC  HL
1D5D 7E        1479           LD   A,(HL)        ;  A = OLD TAIL
1D5E 73        1480           LD   (HL),E        ;  SET TAIL = .OBJ
1D5F A7        1481           AND  A             ;  WAS OLD TAIL NIL?
1D60 2806      1482           JR   Z,ADDTQ1-$    ;  JUMP IF SO
               1483  ;  NONNIL OLD TAIL, SET NEXT(OLDTAIL)=.OBJ
1D62 5F        1484           LD   E,A           ;  DE = .NEXT(OLDTAIL)
1D63 7E        1485           LD   A,(HL)        ;  A = .OBJ (FROM NEW TAIL)
1D64 2B        1486           DEC  HL
1D65 1B        1487           DEC  DE
1D66 12        1488           LD   (DE),A
1D67 C9        1489           RET
               1490  ;  NIL OLD TAIL CASE
1D68 2B        1491  ADDTQ1:  DEC  HL            ;  BACKUP TO HEAD
1D69 73        1492           LD   (HL),E        ;  HEAD = .OBJ
1D6A C9        1493           RET
               1494  ;  SUBROUTINE TO POINT IX AT FIRST ENTRY ON A Q
               1495  ;  ENTRY:         HL = Q HEAD-TAIL
               1496  ;  EXIT:          IX,DE = OBJECT, A = L.O. BYTE OF OBJECT
               1497  ;                 NONZERO STATUS SET IF Q NOT EMPTY
1D6B F3        1498  FIRST:   DI
1D6C 5E        1499           LD   E,(HL)
1D6D 23        1500           INC  HL
1D6E 23        1501           INC  HL
1D6F 56        1502           LD   D,(HL)        ;  D = H.O. ADDR. BYTE
1D70 2B        1503           DEC  HL
1D71 2B        1504           DEC  HL
1D72 7B        1505           LD   A,E           ;  E = HEAD OF Q
1D73 A7        1506           AND  A
1D74 D5        1507           PUSH DE
1D75 DDE1      1508           POP  IX
1D77 C9        1509           RET



               1511  ;  **********************
               1512  ;  * GUNFIGHT CONSTANTS *
               1513  ;  **********************
```

```
ADDR OBJECT    STMT  LABEL   OPCD OPERAND      COMMENT

               1514          ORG  ($+1).AND.0FFFEH
1D78           1515  INTTBL:
1D78 C51B      1516  LFRVEC: DEFW GFLFR
1D7A 5D1B      1517  WRTVEC: DEFW GFWRIT
               1518  ; WAGON LIMITS TABLE
1D7C 0A        1519  WAGLMT: DEFB TLINE
1D7D 44        1520          DEFB BLINE-24
1D7E 47455420  1521  GETRDY: DEFM 'GET READY'
               1522  ; GUNFIGHTER LIMITS
1D87 00        1523  GUNLMT: DEFB 0
1D88 2F        1524          DEFB LCACX-17
1D89 0A        1525          DEFB TLINE
1D8A 48        1526          DEFB BLINE-20
1D8B 44524157  1527  DRAW:   DEFM 'DRAW'
               1528  ; BULLET LIMITS
1D8F 00        1529  BULLMT  DEFB 0
1D90 9F        1530          DEFB 159
1D91 09        1531          DEFB ALINE
1D92 5B        1532          DEFB BLINE-1
               1533  BN      MACR #DX,#ARMX,#DY,#ARMY
               1534          DEFW #DX
               1535          DEFB #ARMX
               1536          DEFW #DY
               1537          DEFB #ARMY
               1538          ENDM
1D93           1539  BULTAB  BN   768,15,768,15
1D93 0003      1539 +        DEFW 768
1D95 0F        1539 +        DEFB 15
1D96 0003      1539 +        DEFW 768
1D98 0F        1539 +        DEFB 15
1D99           1540          BN   1024,15,512,12
1D99 0004      1540 +        DEFW 1024
1D9B 0F        1540 +        DEFB 15
1D9C 0002      1540 +        DEFW 512
1D9E 0C        1540 +        DEFB 12
1D9F           1541          BN   1024,15,256,11
1D9F 0004      1541 +        DEFW 1024
1DA1 0F        1541 +        DEFB 15
1DA2 0001      1541 +        DEFW 256
1DA4 0B        1541 +        DEFB 11
1DA5           1542          BN   1024,15,0,8
1DA5 0004      1542 +        DEFW 1024
1DA7 0F        1542 +        DEFB 15
1DA8 0000      1542 +        DEFW 0
1DAA 08        1542 +        DEFB 8
1DAB           1543          BN   1024,15,-256,6
1DAB 0004      1543 +        DEFW 1024
1DAD 0F        1543 +        DEFB 15
1DAE 00FF      1543 +        DEFW -256
1DB0 06        1543 +        DEFB 6
1DB1           1544          BN   1024,15,-512,4
1DB1 0004      1544 +        DEFW 1024
1DB3 0F        1544 +        DEFB 15
1DB4 00FE      1544 +        DEFW -512
1DB6 04        1544 +        DEFB 4
1DB7           1545          BN   768,15,-768,3
1DB7 0003      1545 +        DEFW 768
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND        COMMENT

1DB9 0F        1545 +         DEFB 15
1DBA 00FD      1545 +         DEFW -768
1DBC 03        1545 +         DEFB 3
1DBD           1546  LFTAB:   DEF5 72,22,44,67,14
1DBD 48        1546 +         DEFB 72
1DBE 16        1546 +         DEFB 22
1DBF 2C        1546 +         DEFB 44
1DC0 43        1546 +         DEFB 67
1DC1 0E        1546 +         DEFB 14
1DC2           1547  RFTAB:   DEF5 18,68,40,13,63
1DC2 12        1547 +         DEFB 18
1DC3 44        1547 +         DEFB 68
1DC4 28        1547 +         DEFB 40
1DC5 0D        1547 +         DEFB 13
1DC6 3F        1547 +         DEFB 63
1DC7 9D        1548  GFCOLS:  DEFB 9DH
1DC8 76        1549           DEFB 76H
1DC9 FC        1550           DEFB 0FCH
1DCA 87        1551           DEFB 87H
1DCB 9D        1552           DEFB 9DH
1DCC 76        1553           DEFB 76H
1DCD 6C        1554           DEFB 6CH
1DCE 87        1555           DEFB 87H
1DCF           1556  SINIT:   DEF8 6,6,0,0,0,30H,30H,0
1DCF 06        1556 +         DEFB 6
1DD0 06        1556 +         DEFB 6
1DD1 00        1556 +         DEFB 0
1DD2 00        1556 +         DEFB 0
1DD3 00        1556 +         DEFB 0
1DD4 30        1556 +         DEFB 30H
1DD5 30        1556 +         DEFB 30H
1DD6 00        1556 +         DEFB 0
1DD7           1557           DEF4 0,80H,0FH,0FH
1DD7 00        1557 +         DEFB 0
1DD8 80        1557 +         DEFB 80H
1DD9 0F        1557 +         DEFB 0FH
1DDA 0F        1557 +         DEFB 0FH
>0007          1558  NUMB:    EQU  00000111B      ; COLOR MASK
>000B          1559  BULT     EQU  00001011B
>000B          1560  TIME     EQU  00001011B
>000B          1561  LARGE:   EQU  00001011B
>000C          1562  LARG2    EQU  00001100B


               1564   ; **********************
               1565   ; * GUN FIGHT PATTERNS *
               1566   ; **********************
               1567   ;
               1568   ; PATTERN TABLES:
1DDB FC1D      1569  ARMTBL:  DEFW ARM0
1DDD 0A1E      1570           DEFW ARM1
1DDF 141E      1571           DEFW ARM2
1DE1 1C1E      1572           DEFW ARM3
1DE3 281E      1573           DEFW ARM4
1DE5 361E      1574           DEFW ARM5
1DE7 461E      1575           DEFW ARM6
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

               1576  ; PATTERN DEFINITION MACROS
               1577  DEF02    MACR #A, #B
               1578           DEFB 0#AH
               1579           DEFB 0#BH
               1580           ENDM
               1581  DEF03    MACR #A, #B, #C
               1582           DEFB 0#AH
               1583           DEFB 0#BH
               1584           DEFB 0#CH
               1585           ENDM
               1586  DEF04    MACR #A, #B, #C, #D
               1587           DEFB 0#AH
               1588           DEFB 0#BH
               1589           DEFB 0#CH
               1590           DEFB 0#DH
               1591           ENDM
1DE9           1592  TREE     DEF2 1, 17
1DE9 01        1592 +         DEFB 1
1DEA 11        1592 +         DEFB 17
1DEB 08        1593           DEFB 00001000B
1DEC 1C        1594           DEFB 00011100B
1DED 3E        1595           DEFB 00111110B
1DEE 6B        1596           DEFB 01101011B
1DEF 08        1597           DEFB 00001000B
1DF0 08        1598           DEFB 00001000B
1DF1 3C        1599           DEFB 00111100B
1DF2 7E        1600           DEFB 01111110B
1DF3 A9        1601           DEFB 10101001B
1DF4 08        1602           DEFB 00001000B
1DF5 3C        1603           DEFB 00111100B
1DF6 7E        1604           DEFB 01111110B
1DF7 EB        1605           DEFB 11101011B
1DF8 89        1606           DEFB 10001001B
1DF9 08        1607           DEFB 00001000B
1DFA 1C        1608           DEFB 00011100B
1DFB AE        1609           DEFB 10101110B
1DFC           1610  ARM0:    DEF04 0A, 0A, 2, 5
1DFC 0A        1610 +         DEFB 00AH
1DFD 0A        1610 +         DEFB 00AH
1DFE 02        1610 +         DEFB 02H
1DFF 05        1610 +         DEFB 05H
1E00           1611           DEF02 40, 00,
1E00 40        1611 +         DEFB 040H
1E01 00        1611 +         DEFB 000H
1E02           1612           DEF02 51, 00,
1E02 51        1612 +         DEFB 051H
1E03 00        1612 +         DEFB 000H
1E04           1613           DEF02 04, 00,
1E04 04        1613 +         DEFB 004H
1E05 00        1613 +         DEFB 000H
1E06           1614           DEF02 01, 00,
1E06 01        1614 +         DEFB 001H
1E07 00        1614 +         DEFB 000H
1E08           1615           DEF02 00, 40,
1E08 00        1615 +         DEFB 000H
1E09 40        1615 +         DEFB 040H
1E0A           1616  ARM1:    DEF04 0A, 0A, 2, 3
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

1E0A 0A        1616 +       DEFB 00AH
1E0B 0A        1616 +       DEFB 00AH
1E0C 02        1616 +       DEFB 02H
1E0D 03        1616 +       DEFB 03H
1E0E           1617         DEF02 50,00,
1E0E 50        1617 +       DEFB 050H
1E0F 00        1617 +       DEFB 000H
1E10           1618         DEF02 14,00,
1E10 14        1618 +       DEFB 014H
1E11 00        1618 +       DEFB 000H
1E12           1619         DEF02 01,40,
1E12 01        1619 +       DEFB 001H
1E13 40        1619 +       DEFB 040H
1E14           1620  ARM2:  DEF04 0A,0A,2,2
1E14 0A        1620 +       DEFB 00AH
1E15 0A        1620 +       DEFB 00AH
1E16 02        1620 +       DEFB 02H
1E17 02        1620 +       DEFB 02H
1E18           1621         DEF02 54,00,
1E18 54        1621 +       DEFB 054H
1E19 00        1621 +       DEFB 000H
1E1A           1622         DEF02 55,40,
1E1A 55        1622 +       DEFB 055H
1E1B 40        1622 +       DEFB 040H
1E1C           1623  ARM3:  DEF04 0A,7,2,4
1E1C 0A        1623 +       DEFB 00AH
1E1D 07        1623 +       DEFB 07H
1E1E 02        1623 +       DEFB 02H
1E1F 04        1623 +       DEFB 04H
1E20           1624         DEF02 10,00,
1E20 10        1624 +       DEFB 010H
1E21 00        1624 +       DEFB 000H
1E22           1625         DEF02 05,40,
1E22 05        1625 +       DEFB 005H
1E23 40        1625 +       DEFB 040H
1E24           1626         DEF02 54,00,
1E24 54        1626 +       DEFB 054H
1E25 00        1626 +       DEFB 000H
1E26           1627         DEF02 50,00,
1E26 50        1627 +       DEFB 050H
1E27 00        1627 +       DEFB 000H
1E28           1628  ARM4:  DEF04 0A,6,2,5
1E28 0A        1628 +       DEFB 00AH
1E29 06        1628 +       DEFB 06H
1E2A 02        1628 +       DEFB 02H
1E2B 05        1628 +       DEFB 05H
1E2C           1629         DEF02 00,40,
1E2C 00        1629 +       DEFB 000H
1E2D 40        1629 +       DEFB 040H
1E2E           1630         DEF02 45,00,
1E2E 45        1630 +       DEFB 045H
1E2F 00        1630 +       DEFB 000H
1E30           1631         DEF02 10,00,
1E30 10        1631 +       DEFB 010H
1E31 00        1631 +       DEFB 000H
1E32           1632         DEF02 50,00,
1E32 50        1632 +       DEFB 050H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1E33 00        1632 +         DEFB 000H
1E34           1633           DEF02 40,00,
1E34 40        1633 +         DEFB 040H
1E35 00        1633 +         DEFB 000H
1E36           1634  ARM5:    DEF04 0A,5,2,6
1E36 0A        1634 +         DEFB 00AH
1E37 05        1634 +         DEFB 05H
1E38 02        1634 +         DEFB 02H
1E39 06        1634 +         DEFB 06H
1E3A           1635           DEF02 00,40,
1E3A 00        1635 +         DEFB 000H
1E3B 40        1635 +         DEFB 040H
1E3C           1636           DEF02 01,00,
1E3C 01        1636 +         DEFB 001H
1E3D 00        1636 +         DEFB 000H
1E3E           1637           DEF02 05,00,
1E3E 05        1637 +         DEFB 005H
1E3F 00        1637 +         DEFB 000H
1E40           1638           DEF02 14,00,
1E40 14        1638 +         DEFB 014H
1E41 00        1638 +         DEFB 000H
1E42           1639           DEF02 54,00,
1E42 54        1639 +         DEFB 054H
1E43 00        1639 +         DEFB 000H
1E44           1640           DEF02 50,00,
1E44 50        1640 +         DEFB 050H
1E45 00        1640 +         DEFB 000H
1E46           1641  ARM6:    DEF04 0A,5,1,5
1E46 0A        1641 +         DEFB 00AH
1E47 05        1641 +         DEFB 05H
1E48 01        1641 +         DEFB 01H
1E49 05        1641 +         DEFB 05H
1E4A 01        1642           DEFB 01H
1E4B 44        1643           DEFB 44H
1E4C 10        1644           DEFB 10H
1E4D 40        1645           DEFB 40H
1E4E 40        1646           DEFB 40H
               1647  ; **** NOTE ****
               1648  ; THE FOLLOWING PATTERNS ARE CONSTRAINED TO EXIST ON THE
               1649  ; PAGE.  THE FOLLOWING 'ORG' WILL DO IT FOR EXPERIMENTAL
               1650  ; PATTERNS ARE: LEG0,LEG1,LEG2,KIL1,KIL2
               1651  ;        ORG     ($+255).AND.0FF00H      ;
1E4F 64        1652  LEG0:    DEFB LEG1.AND.0FFH
1E50 04        1653           DEFB 4
1E51           1654           DEF04 0,0F,3,5
1E51 00        1654 +         DEFB 00H
1E52 0F        1654 +         DEFB 00FH
1E53 03        1654 +         DEFB 03H
1E54 05        1654 +         DEFB 05H
1E55           1655           DEF03 01,55,00,
1E55 01        1655 +         DEFB 001H
1E56 55        1655 +         DEFB 055H
1E57 00        1655 +         DEFB 000H
1E58           1656           DEF03 05,45,40,
1E58 05        1656 +         DEFB 005H
1E59 45        1656 +         DEFB 045H
1E5A 40        1656 +         DEFB 040H
```

```
ADDR OBJECT   STMT  LABEL   OPCD OPERAND      COMMENT

1E5B          1657              DEF03 15,01,40,
1E5B 15       1657 +        DEFB 015H
1E5C 01       1657 +        DEFB 001H
1E5D 40       1657 +        DEFB 040H
1E5E          1658          DEF03 50,01,40,
1E5E 50       1658 +        DEFB 050H
1E5F 01       1658 +        DEFB 001H
1E60 40       1658 +        DEFB 040H
1E61          1659          DEF03 15,00,54,
1E61 15       1659 +        DEFB 015H
1E62 00       1659 +        DEFB 000H
1E63 54       1659 +        DEFB 054H
1E64 74       1660  LEG1:   DEFB LEG2.AND.0FFH
1E65 04       1661          DEFB 4
1E66          1662          DEF04 2,0F,2,5
1E66 02       1662 +        DEFB 02H
1E67 0F       1662 +        DEFB 00FH
1E68 02       1662 +        DEFB 02H
1E69 05       1662 +        DEFB 05H
1E6A          1663          DEF02 15,50,
1E6A 15       1663 +        DEFB 015H
1E6B 50       1663 +        DEFB 050H
1E6C          1664          DEF02 54,50,
1E6C 54       1664 +        DEFB 054H
1E6D 50       1664 +        DEFB 050H
1E6E          1665          DEF02 50,50,
1E6E 50       1665 +        DEFB 050H
1E6F 50       1665 +        DEFB 050H
1E70          1666          DEF02 50,50,
1E70 50       1666 +        DEFB 050H
1E71 50       1666 +        DEFB 050H
1E72          1667          DEF02 55,15,
1E72 55       1667 +        DEFB 055H
1E73 15       1667 +        DEFB 015H
1E74 4F       1668  LEG2:   DEFB LEG0.AND.0FFH
1E75 04       1669          DEFB 4
1E76          1670          DEF04 3,0F,2,5
1E76 03       1670 +        DEFB 03H
1E77 0F       1670 +        DEFB 00FH
1E78 02       1670 +        DEFB 02H
1E79 05       1670 +        DEFB 05H
1E7A          1671          DEF02 55,00,
1E7A 55       1671 +        DEFB 055H
1E7B 00       1671 +        DEFB 000H
1E7C          1672          DEF02 15,00,
1E7C 15       1672 +        DEFB 015H
1E7D 00       1672 +        DEFB 000H
1E7E          1673          DEF02 15,00,
1E7E 15       1673 +        DEFB 015H
1E7F 00       1673 +        DEFB 000H
1E80          1674          DEF02 14,00,
1E80 14       1674 +        DEFB 014H
1E81 00       1674 +        DEFB 000H
1E82          1675          DEF02 05,40,
1E82 05       1675 +        DEFB 005H
1E83 40       1675 +        DEFB 040H
1E84 D6       1676  KIL1:   DEFB KIL2.AND.0FFH
```

```
1E85 14        1677.           DEFB 20
1E86           1678            DEF04 0,1,4,13
1E86 00        1678 +          DEFB 00H
1E87 01        1678 +          DEFB 01H
1E88 04        1678 +          DEFB 04H
1E89 13        1678 +          DEFB 013H
1E8A           1679            DEF04 01,10,00,00,
1E8A 01        1679 +          DEFB 001H
1E8B 10        1679 +          DEFB 010H
1E8C 00        1679 +          DEFB 000H
1E8D 00        1679 +          DEFB 000H
1E8E           1680            DEF04 45,54,40,00,
1E8E 45        1680 +          DEFB 045H
1E8F 54        1680 +          DEFB 054H
1E90 40        1680 +          DEFB 040H
1E91 00        1680 +          DEFB 000H
1E92           1681            DEF04 55,55,40,00,
1E92 55        1681 +          DEFB 055H
1E93 55        1681 +          DEFB 055H
1E94 40        1681 +          DEFB 040H
1E95 00        1681 +          DEFB 000H
1E96           1682            DEF04 0A,A8,00,00,
1E96 0A        1682 +          DEFB 00AH
1E97 A8        1682 +          DEFB 0A8H
1E98 00        1682 +          DEFB 000H
1E99 00        1682 +          DEFB 000H
1E9A           1683            DEF04 0A,A2,00,01,
1E9A 0A        1683 +          DEFB 00AH
1E9B A2        1683 +          DEFB 0A2H
1E9C 00        1683 +          DEFB 000H
1E9D 01        1683 +          DEFB 001H
1E9E           1684            DEF04 0A,AA,80,14,
1E9E 0A        1684 +          DEFB 00AH
1E9F AA        1684 +          DEFB 0AAH
1EA0 80        1684 +          DEFB 080H
1EA1 14        1684 +          DEFB 014H
1EA2           1685            DEF04 02,AA,00,50,
1EA2 02        1685 +          DEFB 002H
1EA3 AA        1685 +          DEFB 0AAH
1EA4 00        1685 +          DEFB 000H
1EA5 50        1685 +          DEFB 050H
1EA6           1686            DEF04 00,A8,05,40,
1EA6 00        1686 +          DEFB 000H
1EA7 A8        1686 +          DEFB 0A8H
1EA8 05        1686 +          DEFB 005H
1EA9 40        1686 +          DEFB 040H
1EAA           1687            DEF04 05,55,54,00,
1EAA 05        1687 +          DEFB 005H
1EAB 55        1687 +          DEFB 055H
1EAC 54        1687 +          DEFB 054H
1EAD 00        1687 +          DEFB 000H
1EAE           1688            DEF04 15,55,50,00,
1EAE 15        1688 +          DEFB 015H
1EAF 55        1688 +          DEFB 055H
1EB0 50        1688 +          DEFB 050H
1EB1 00        1688 +          DEFB 000H
1EB2           1689            DEF04 54,55,50,00,
ADDR OBJECT    STMT  LABEL   OPCD OPERAND       COMMENT

1EB2 54        1689 +        DEFB 054H
1EB3 55        1689 +        DEFB 055H
1EB4 50        1689 +        DEFB 050H
1EB5 00        1689 +        DEFB 000H
1EB6           1690          DEF04 50,05,54,00,
1EB6 50        1690 +        DEFB 050H
1EB7 05        1690 +        DEFB 005H
1EB8 54        1690 +        DEFB 054H
1EB9 00        1690 +        DEFB 000H
1EBA           1691          DEF04 50,01,55,00,
1EBA 50        1691 +        DEFB 050H
1EBB 01        1691 +        DEFB 001H
1EBC 55        1691 +        DEFB 055H
1EBD 00        1691 +        DEFB 000H
1EBE           1692          DEF04 10,01,55,40,
1EBE 10        1692 +        DEFB 010H
1EBF 01        1692 +        DEFB 001H
1EC0 55        1692 +        DEFB 055H
1EC1 40        1692 +        DEFB 040H
1EC2           1693          DEF04 10,00,05,50,
1EC2 10        1693 +        DEFB 010H
1EC3 00        1693 +        DEFB 000H
1EC4 05        1693 +        DEFB 005H
1EC5 50        1693 +        DEFB 050H
1EC6           1694          DEF04 00,00,01,50,
1EC6 00        1694 +        DEFB 000H
1EC7 00        1694 +        DEFB 000H
1EC8 01        1694 +        DEFB 001H
1EC9 50        1694 +        DEFB 050H
1ECA           1695          DEF04 00,00,00,40,
1ECA 00        1695 +        DEFB 000H
1ECB 00        1695 +        DEFB 000H
1ECC 00        1695 +        DEFB 000H
1ECD 40        1695 +        DEFB 040H
1ECE           1696          DEF04 00,00,01,40,
1ECE 00        1696 +        DEFB 000H
1ECF 00        1696 +        DEFB 000H
1ED0 01        1696 +        DEFB 001H
1ED1 40        1696 +        DEFB 040H
1ED2           1697          DEF04 00,00,00,54,
1ED2 00        1697 +        DEFB 000H
1ED3 00        1697 +        DEFB 000H
1ED4 00        1697 +        DEFB 000H
1ED5 54        1697 +        DEFB 054H
1ED6 D6        1698  KIL2:   DEFB KIL2.AND.0FFH
1ED7 3C        1699          DEFB 60
1ED8           1700          DEF04 0,D,4,7
1ED8 00        1700 +        DEFB 00H
1ED9 0D        1700 +        DEFB 0DH
1EDA 04        1700 +        DEFB 04H
1EDB 07        1700 +        DEFB 07H
1EDC           1701          DEF04 01,10,00,00,
1EDC 01        1701 +        DEFB 001H
1EDD 10        1701 +        DEFB 010H
1EDE 00        1701 +        DEFB 000H
1EDF 00        1701 +        DEFB 000H
1EE0           1702          DEF04 45,54,40,00,
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1EE0 45        1702 +         DEFB 045H
1EE1 54        1702 +         DEFB 054H
1EE2 40        1702 +         DEFB 040H
1EE3 00        1702 +         DEFB 000H
1EE4           1703           DEF04 55,55,40,00,
1EE4 55        1703 +         DEFB 055H
1EE5 55        1703 +         DEFB 055H
1EE6 40        1703 +         DEFB 040H
1EE7 00        1703 +         DEFB 000H
1EE8           1704           DEF04 0A,A8,00,00,
1EE8 0A        1704 +         DEFB 00AH
1EE9 A8        1704 +         DEFB 0A8H
1EEA 00        1704 +         DEFB 000H
1EEB 00        1704 +         DEFB 000H
1EEC           1705           DEF04 0A,88,15,01,
1EEC 0A        1705 +         DEFB 00AH
1EED 88        1705 +         DEFB 088H
1EEE 15        1705 +         DEFB 015H
1EEF 01        1705 +         DEFB 001H
1EF0           1706           DEF04 16,A5,55,41,
1EF0 16        1706 +         DEFB 016H
1EF1 A5        1706 +         DEFB 0A5H
1EF2 55        1706 +         DEFB 055H
1EF3 41        1706 +         DEFB 041H
1EF4           1707           DEF04 15,55,55,55,
1EF4 15        1707 +         DEFB 015H
1EF5 55        1707 +         DEFB 055H
1EF6 55        1707 +         DEFB 055H
1EF7 55        1707 +         DEFB 055H
1EF8           1708  CACTUS   DEF2 1,12
1EF8 01        1708 +         DEFB 1
1EF9 0C        1708 +         DEFB 12
1EFA 20        1709           DEFB 00100000B
1EFB 30        1710           DEFB 00110000B
1EFC 38        1711           DEFB 00111000B
1EFD 30        1712           DEFB 00110000B
1EFE B2        1713           DEFB 10110010B
1EFF F2        1714           DEFB 11110010B
1F00 F6        1715           DEFB 11110110B
1F01 3C        1716           DEFB 00111100B
1F02 3C        1717           DEFB 00111100B
1F03 30        1718           DEFB 00110000B
1F04 30        1719           DEFB 00110000B
1F05 30        1720           DEFB 00110000B
1F06 474F5420  1721  GOTME:   DEFM 'GOT ME'
1F0C 00        1722  NULPAT:  DEFB 0
1F0D 00        1723           DEFB 0
1F0E 01        1724           DEFB 1
1F0F 01        1725           DEFB 1
1F10           1726  GFBODY:  DEF04 0,0,3,F
1F10 00        1726 +         DEFB 00H
1F11 00        1726 +         DEFB 00H
1F12 03        1726 +         DEFB 03H
1F13 0F        1726 +         DEFB 0FH
1F14           1727           DEF03 00,44,00,
1F14 00        1727 +         DEFB 000H
1F15 44        1727 +         DEFB 044H
```

```
ADDR OBJECT   STMT   LABEL   OPCD OPERAND      COMMENT

1F16 00       1727 +         DEFB 000H
1F17          1728           DEF03 11,55,10,
1F17 11       1728 +         DEFB 011H
1F18 55       1728 +         DEFB 055H
1F19 10       1728 +         DEFB 010H
1F1A          1729           DEF03 15,55,50,
1F1A 15       1729 +         DEFB 015H
1F1B 55       1729 +         DEFB 055H
1F1C 50       1729 +         DEFB 050H
1F1D          1730           DEF03 02,AA,00,
1F1D 02       1730 +         DEFB 002H
1F1E AA       1730 +         DEFB 0AAH
1F1F 00       1730 +         DEFB 000H
1F20          1731           DEF03 02,A2,00,
1F20 02       1731 +         DEFB 002H
1F21 A2       1731 +         DEFB 0A2H
1F22 00       1731 +         DEFB 000H
1F23          1732           DEF03 02,AA,80,
1F23 02       1732 +         DEFB 002H
1F24 AA       1732 +         DEFB 0AAH
1F25 80       1732 +         DEFB 080H
1F26          1733           DEF03 00,AA,00,
1F26 00       1733 +         DEFB 000H
1F27 AA       1733 +         DEFB 0AAH
1F28 00       1733 +         DEFB 000H
1F29          1734           DEF03 00,A8,00,
1F29 00       1734 +         DEFB 000H
1F2A A8       1734 +         DEFB 0A8H
1F2B 00       1734 +         DEFB 000H
1F2C          1735           DEF03 15,55,00,
1F2C 15       1735 +         DEFB 015H
1F2D 55       1735 +         DEFB 055H
1F2E 00       1735 +         DEFB 000H
1F2F          1736           DEF03 55,55,50,
1F2F 55       1736 +         DEFB 055H
1F30 55       1736 +         DEFB 055H
1F31 50       1736 +         DEFB 050H
1F32          1737           DEF03 51,55,50,
1F32 51       1737 +         DEFB 051H
1F33 55       1737 +         DEFB 055H
1F34 50       1737 +         DEFB 050H
1F35          1738           DEF03 41,55,00,
1F35 41       1738 +         DEFB 041H
1F36 55       1738 +         DEFB 055H
1F37 00       1738 +         DEFB 000H
1F38          1739           DEF03 41,55,00,
1F38 41       1739 +         DEFB 041H
1F39 55       1739 +         DEFB 055H
1F3A 00       1739 +         DEFB 000H
1F3B          1740           DEF03 45,55,00,
1F3B 45       1740 +         DEFB 045H
1F3C 55       1740 +         DEFB 055H
1F3D 00       1740 +         DEFB 000H
1F3E 01       1741           DEFB 01H
1F3F 55       1742           DEFB 55H
1F40          1743   WAGPAT: DEF04 0,0,4,16
1F40 00       1743 +         DEFB 00H
```

```
ADDR OBJECT    STMT  LABEL    OPCD OPERAND       COMMENT

1F41 00        1743 +         DEFB 00H
1F42 04        1743 +         DEFB 04H
1F43 16        1743 +         DEFB 016H
1F44           1744           DEF04 00,05,50,00,
1F44 00        1744 +         DEFB 000H
1F45 05        1744 +         DEFB 005H
1F46 50        1744 +         DEFB 050H
1F47 00        1744 +         DEFB 000H
1F48           1745           DEF04 00,55,55,00,
1F48 00        1745 +         DEFB 000H
1F49 55        1745 +         DEFB 055H
1F4A 55        1745 +         DEFB 055H
1F4B 00        1745 +         DEFB 000H
1F4C           1746           DEF04 01,55,55,40,
1F4C 01        1746 +         DEFB 001H
1F4D 55        1746 +         DEFB 055H
1F4E 55        1746 +         DEFB 055H
1F4F 40        1746 +         DEFB 040H
1F50           1747           DEF04 05,55,55,50,
1F50 05        1747 +         DEFB 005H
1F51 55        1747 +         DEFB 055H
1F52 55        1747 +         DEFB 055H
1F53 50        1747 +         DEFB 050H
1F54           1748           DEF04 15,54,15,54,
1F54 15        1748 +         DEFB 015H
1F55 54        1748 +         DEFB 054H
1F56 15        1748 +         DEFB 015H
1F57 54        1748 +         DEFB 054H
1F58           1749           DEF04 15,50,05,54,
1F58 15        1749 +         DEFB 015H
1F59 50        1749 +         DEFB 050H
1F5A 05        1749 +         DEFB 005H
1F5B 54        1749 +         DEFB 054H
1F5C           1750           DEF04 15,40,01,54,
1F5C 15        1750 +         DEFB 015H
1F5D 40        1750 +         DEFB 040H
1F5E 01        1750 +         DEFB 001H
1F5F 54        1750 +         DEFB 054H
1F60           1751           DEF04 15,40,01,54,
1F60 15        1751 +         DEFB 015H
1F61 40        1751 +         DEFB 040H
1F62 01        1751 +         DEFB 001H
1F63 54        1751 +         DEFB 054H
1F64           1752           DEF04 15,50,05,54,
1F64 15        1752 +         DEFB 015H
1F65 50        1752 +         DEFB 050H
1F66 05        1752 +         DEFB 005H
1F67 54        1752 +         DEFB 054H
1F68           1753           DEF04 05,54,15,50,
1F68 05        1753 +         DEFB 005H
1F69 54        1753 +         DEFB 054H
1F6A 15        1753 +         DEFB 015H
1F6B 50        1753 +         DEFB 050H
1F6C           1754           DEF04 01,55,55,40,
1F6C 01        1754 +         DEFB 001H
1F6D 55        1754 +         DEFB 055H
1F6E 55        1754 +         DEFB 055H
```

```
1F6F 40       1754 +        DEFB 040H
1F70          1755          DEF04 00,55,55,00,
1F70 00       1755 +        DEFB 000H
1F71 55       1755 +        DEFB 055H
1F72 55       1755 +        DEFB 055H
1F73 00       1755 +        DEFB 000H
1F74          1756          DEF04 00,15,54,00,
1F74 00       1756 +        DEFB 000H
1F75 15       1756 +        DEFB 015H
1F76 54       1756 +        DEFB 054H
1F77 00       1756 +        DEFB 000H
1F78          1757          DEF04 02,AA,AA,80,
1F78 02       1757 +        DEFB 002H
1F79 AA       1757 +        DEFB 0AAH
1F7A AA       1757 +        DEFB 0AAH
1F7B 80       1757 +        DEFB 080H
1F7C          1758          DEF04 00,AA,AA,00,
1F7C 00       1758 +        DEFB 000H
1F7D AA       1758 +        DEFB 0AAH
1F7E AA       1758 +        DEFB 0AAH
1F7F 00       1758 +        DEFB 000H
1F80          1759          DEF04 12,AA,AA,84,
1F80 12       1759 +        DEFB 012H
1F81 AA       1759 +        DEFB 0AAH
1F82 AA       1759 +        DEFB 0AAH
1F83 84       1759 +        DEFB 084H
1F84          1760          DEF04 10,A8,2A,04,
1F84 10       1760 +        DEFB 010H
1F85 A8       1760 +        DEFB 0A8H
1F86 2A       1760 +        DEFB 02AH
1F87 04       1760 +        DEFB 004H
1F88          1761          DEF04 10,20,08,04,
1F88 10       1761 +        DEFB 010H
1F89 20       1761 +        DEFB 020H
1F8A 08       1761 +        DEFB 008H
1F8B 04       1761 +        DEFB 004H
1F8C          1762          DEF04 52,AA,AA,85,
1F8C 52       1762 +        DEFB 052H
1F8D AA       1762 +        DEFB 0AAH
1F8E AA       1762 +        DEFB 0AAH
1F8F 85       1762 +        DEFB 085H
1F90          1763          DEF04 10,20,08,04,
1F90 10       1763 +        DEFB 010H
1F91 20       1763 +        DEFB 020H
1F92 08       1763 +        DEFB 008H
1F93 04       1763 +        DEFB 004H
1F94          1764          DEF04 10,00,00,04,
1F94 10       1764 +        DEFB 010H
1F95 00       1764 +        DEFB 000H
1F96 00       1764 +        DEFB 000H
1F97 04       1764 +        DEFB 004H
1F98          1765          DEF04 10,00,00,04,
1F98 10       1765 +        DEFB 010H
1F99 00       1765 +        DEFB 000H
1F9A 00       1765 +        DEFB 000H
1F9B 04       1765 +        DEFB 004H
              1766  ;
```

```
1F9C 00        1767.  FUDG4:   DEFB 0
               1768   ;
1F9D           1769   MSET     MASTER 0A4
1F9D 80        1769 +          DEFB 80H
1F9E 11        1769 +          DEFB 0A4
1F9F           1770            VOLUME 09H,0H
1F9F B0        1770 +          DEFB 0B0H
1FA0 09        1770 +          DEFB 09H
1FA1 00        1770 +          DEFB 0H
1FA2 C9        1771            RET
               1772     ; HOME ON THE RANGE
1FA3 CD9D1F    1773   HOME     CALL MSET
1FA6           1774            NOTE1 36,G1
1FA6 24        1774 +          DEFB 36&7FH
1FA7 7E        1774 +          DEFB G1
1FA8           1775            NOTE1 12,F1
1FA8 0C        1775 +          DEFB 12&7FH
1FA9 8D        1775 +          DEFB F1
1FAA           1776            NOTE1 18,E1
1FAA 12        1776 +          DEFB 18&7FH
1FAB 96        1776 +          DEFB E1
1FAC           1777            NOTE1 6,D1
1FAC 06        1777 +          DEFB 6&7FH
1FAD A8        1777 +          DEFB D1
1FAE           1778            NOTE1 36,E1
1FAE 24        1778 +          DEFB 36&7FH
1FAF 96        1778 +          DEFB E1
1FB0           1779            QUIET
1FB0 F0        1779 +          DEFB 0F0H
               1780     ; TAPS
1FB1           1781   TAPS
1FB1 CD9D1F    1782            CALL MSET
1FB4           1783            NOTE1 18,C1
1FB4 12        1783 +          DEFB 18&7FH
1FB5 BD        1783 +          DEFB C1
1FB6           1784            NOTE1 6,C1
1FB6 06        1784 +          DEFB 6&7FH
1FB7 BD        1784 +          DEFB C1
1FB8           1785            NOTE1 36,F1
1FB8 24        1785 +          DEFB 36&7FH
1FB9 8D        1785 +          DEFB F1
1FBA           1786            NOTE1 18,C1
1FBA 12        1786 +          DEFB 18&7FH
1FBB BD        1786 +          DEFB C1
1FBC           1787            NOTE1 6,F1
1FBC 06        1787 +          DEFB 6&7FH
1FBD 8D        1787 +          DEFB F1
1FBE           1788            NOTE1 36,A1
1FBE 24        1788 +          DEFB 36&7FH
1FBF 70        1788 +          DEFB A1
1FC0           1789            QUIET
1FC0 F0        1789 +          DEFB 0F0H
               1790     ; FUNERAL
1FC1           1791   FUNERL
1FC1 CD9D1F    1792            CALL MSET
1FC4           1793            NOTE1 24,A0
1FC4 18        1793 +          DEFB 24&7FH
```

```
1FC5 E1        1793 +          DEFB A0
1FC6           1794            NOTE1  18,A0
1FC6 12        1794 +          DEFB 18&7FH
1FC7 E1        1794 +          DEFB A0
1FC8           1795            NOTE1  6,A0
1FC8 06        1795 +          DEFB 6&7FH
1FC9 E1        1795 +          DEFB A0
1FCA           1796            NOTE1  24,A0
1FCA 18        1796 +          DEFB 24&7FH
1FCB E1        1796 +          DEFB A0
1FCC           1797            NOTE1  18,C1
1FCC 12        1797 +          DEFB 18&7FH
1FCD BD        1797 +          DEFB C1
1FCE           1798            NOTE1  6,B0
1FCE 06        1798 +          DEFB 6&7FH
1FCF C8        1798 +          DEFB B0
1FD0           1799            NOTE1  18,B0
1FD0 12        1799 +          DEFB 18&7FH
1FD1 C8        1799 +          DEFB B0
1FD2           1800            NOTE1  6,A0
1FD2 06        1800 +          DEFB 6&7FH
1FD3 E1        1800 +          DEFB A0
1FD4           1801            NOTE1  18,A0
1FD4 12        1801 +          DEFB 18&7FH
1FD5 E1        1801 +          DEFB A0
1FD6           1802            NOTE1  6,GS0
1FD6 06        1802 +          DEFB 6&7FH
1FD7 EE        1802 +          DEFB GS0
1FD8           1803            NOTE1  18,A0
1FD8 12        1803 +          DEFB 18&7FH
1FD9 E1        1803 +          DEFB A0
1FDA           1804            QUIET
1FDA F0        1804 +          DEFB 0F0H
1FDB           1805   GUNSHOT  OUTPUT  18H,0F0H,0F5H,0FDH,0FFH,0,3FH,0FFH,0EFH
               1805 +          IF    .NOT.(18H=18H)
               1805 +          ENDIF
               1805 +          IF    18H=18H
1FDB 88        1805 +          DEFB 88H
1FDC           1805 +          DEF8 0EFH,0FFH,3FH,0,0FFH,0FDH,0F5H,0F0H
1FDC EF        1805 +          DEFB 0EFH
1FDD FF        1805 +          DEFB 0FFH
1FDE 3F        1805 +          DEFB 3FH
1FDF 00        1805 +          DEFB 0
1FE0 FF        1805 +          DEFB 0FFH
1FE1 FD        1805 +          DEFB 0FDH
1FE2 F5        1805 +          DEFB 0F5H
1FE3 F0        1805 +          DEFB 0F0H
               1805 +          ENDIF
1FE4           1806            LEGSTA
1FE4 E0        1806 +          DEFB 0E0H
1FE5           1807            VOLUME  0FFH,03FH
1FE5 B0        1807 +          DEFB 0B0H
1FE6 FF        1807 +          DEFB 0FFH
1FE7 3F        1807 +          DEFB 03FH
1FE8           1808            REST  5
1FE8 E1        1808 +          DEFB 0E1H
1FE9 05        1808 +          DEFB 5
```

```
 ADDR OBJECT   STMT  LABEL    OPCD OPERAND       COMMENT

 1FEA          1809           NOTE1  5,8FH
 1FEA 05       1809 +         DEFB 5&7FH
 1FEB 8F       1809 +         DEFB 8FH
 1FEC          1810           NOTE1  5,4CH
 1FEC 05       1810 +         DEFB 5&7FH
 1FED 4C       1810 +         DEFB 4CH
 1FEE          1811           QUIET
 1FEE F0       1811 +         DEFB 0F0H
>1FEF          1812  LASTB    EQU  $



               1814  ; *************
               1815  ; * RAM CELLS *
               1816  ; *************
               1817           ORG  NORMEM+0E70H
 4E70          1818           DEFS 150           ; ALLOW BIG STACK
>4F06          1819  STACK    EQU  $             ; START STACK HERE
 4F06          1820           DEFS 12
>4F12          1821  MSTACK   EQU  $
>4F12          1822  STRRAM   EQU  $
 4F12          1823  WRITQ:   DEFS 3             ; WRITE Q HEADER
 4F15          1824  VECQ:    DEFS 3             ; VECTOR Q HEADER
>4F18          1825  VECSTR   EQU  $
 4F18          1826  BULV1:   DEFS BULVSZ        ; BULLET VECTOR 1
 4F2A          1827  BULV2:   DEFS BULVSZ        ; BULLET VECTOR 2
 4F3C          1828  BULV3:   DEFS BULVSZ        ; BULLET VECTOR 3
 4F4E          1829  BULV4:   DEFS BULVSZ        ; BULLET VECTOR 4
 4F60          1830           DEFS 1             ; LEFT COWBOY LINK
 4F61          1831  LCOWB:   DEFS GFVSIZ-1      ; LEFT GUNFIGHTER
 4F77          1832           DEFS 1             ; RIGHT COWBOY LINK
 4F78          1833  RCOWB:   DEFS GFVSIZ-1      ; RIGHT GUNFIGHER
 4F8E          1834           DEFS 1             ; WAGON LINK
 4F8F          1835  WAGVEC:  DEFS WAGVSZ        ; WAGON VECTOR
>4F90          1836  WAGON    EQU  WAGVEC+VBSTAT
>4FA1          1837  ENDRAM   EQU  $
>4FDA          1838  LBULS    EQU  CT5
>4FDB          1839  RBULS    EQU  CT6
 4FA1          1840  RFIELD   DEFS 1
 4FA2          1841  LSCORE   DEFS 3
 4FA5          1842  LFIELD   DEFS 1
 4FA6          1843  RSCORE   DEFS 3
               1844           LIST S
>1FEF          1845  LEND     EQU  LASTB
 4FA9          1846           END

TOTAL ASSEMBLER ERRORS =
                                   CROSS REFERENCE

LABEL      VALUE        REFERENCE

A0          00E1         -508  1794  1795  1796  1797  1801  1802  1804
A1          0070         -520  1789
A2          0037         -532
A3          001B         -544
A4          000D         -556
A5          0006         -562
ACTINT      000E         -225
ADDTQ       1D54        -1355  1110  1274  1423  1441
ADDTQ1      1D68        -1372  1482
ALINE       0009         -676  1053  1055  1055  1531
ALKEYS      0214          -49  1180
ARM0        1DFC        -1479  1569
ARM1        1E0A        -1479  1570
ARM2        1E14        -1479  1571
ARM3        1E1C        -1479  1572
ARM4        1E28        -1479  1573
ARM5        1E36        -1479  1574
ARM6        1E46        -1479  1575
ARMTBL      1DDB        -1439  1255
AS0         00D4         -509
AS1         006A         -521
AS2         0034         -533
AS3         001A         -545
B0          00C8         -510  1799  1800
B1          0064         -522
B2          0031         -534
B3          0018         -546
BCACY       0046         -667   668
BCDADD      0062         -277
BCDCHS      006A         -281
BCDDIV      0068         -280
BCDMUL      0066         -279
BCDNEG      006C         -282
BCDSUB      0064         -278
BEGINT      1B61        -1132
BEGRAM      4FCE         -594
BELP        1859         -773   785
BERASE      183A         -756   757
BITSPL      00A0          -43
BLANK       002A         -243  1148  1169
BLINE       005C         -677  1031  1055  1520  1526  1532
BMUSIC      0012         -229   814   814   937   937  1037
BORG        1AAD        -1064  1111
BOTLIN      0000         -685  1347
BSY         0002         -656   713   771  1063  1069  1155
BTREEY      0041         -668
BULLMT      1D8F        -1410   868  1355
BULLP       1AB9        -1068  1131
BULRIT      1B57        -1125  1159  1165
BULT        000B        -1429  1156
BULTAB      1D93        -1420   792
BULV1       4F18        -1541   728  1121  1186  1291  1304  1353
BULV2       4F2A        -1542
```

```
BULV3     4F3C       -1543   734
BULV4     4F4E       -1544
BULVSZ    0012        -679   744  1120  1187  1292  1336  1356  1826
                      1827  1828  1829
BYTEPL    0028         -42  1053  1055  1055
C1        00BD        -511  1784  1785  1787  1798
C2        005E        -523
C3        002E        -535
C4        0017        -547
C5        000B        -557
C6        0005        -563
C7        0002        -566
CACTUS    1EF8       -1499   949  1118
CACW      19C8        -953   958   964   972   981
CBA       0009        -123
CBB       0007        -121   735
CBC       0006        -120
CBD       0005        -119
CBE       0004        -118
CBFLAG    0008        -122
CBH       000B        -125
CBIXH     0003        -117
CBIXL     0002        -116
CBIYH     0001        -115
CBIYL     0000        -114
CBL       000A        -124
CCACX     004C        -671   877  1115
CHDOWN    0001        -111
CHLEFT    0002        -110
CHRDIS    0032        -248  1154  1164  1221  1222  1223  1224  1225
CHRIGH    0003        -109
CHTRIG    0004        -108
CHUP      0000        -112
CNT       4FDD        -611
COLOL     0004        -168
COLOR     0000        -164
COL1L     0005        -169
COL1R     0001        -165
COL2L     0006        -170
COL2R     0002        -166
COL3L     0007        -171
COL3R     0003        -167
COLBX     000B        -172
COLLST    4FE8        -622
COLSET    0018        -234  1035
CONCM     0008        -189
COWINT    1D34       -1344  1097  1100
COWX      0060        -673
CS1       00B2        -512
CS2       0059        -524
CS3       002C        -536
CS4       0015        -548
CS5       000A        -558
CTO       4FD5        -602
CT1       4FD6        -603
CT2       4FD7        -604
CT3       4FD8        -605
```

```
CT4       4FD9       -606
CT5       4FDA       -607  1048  1838
CT6       4FDB       -608  1839
CT7       4FDC       -609   708   716   758   763  1025
CTIMER    0203        -46
D1        00A8       -513  1778
D2        0054       -525
D3        0029       -537
D4        0014       -549
DABS      0072       -285
DADD      006E       -283
DCLOCK    17E1       -704  1219
DCOUT     17F7       -713   710
DEATH     1B10      -1113
DECCTS    0010       -226   706   706
DELQ      1D29      -1335  1239  1370  1438
DIE       1930       -881   895
DIE1      194C       -892   909
DIE4      1963       -900   922
DIFER     4FFF      -1213
DISNUM    0036       -250   712   712  1062  1068
DISTIM    0052       -267
DLEFT     1942       -888   902
DOIT      0044       -260  1182
DOITB     0046       -261
DRAW      1D8B      -1408  1153
DRX       0040       -663  1153
DS1       009F       -514
DS2       004F       -526
DS3       0027       -538
DS4       0013       -550
DS5       0009       -559
DS6       0004       -564
DSMG      0070       -284
DTAB      1B38      -1133  1182
DURAT     4FEA       -624
E1        0096       -515  1777  1779
E2        004A       -527
E3        0025       -539
E4        0012       -551
EIRE      1BBE      -1167
ELOP      1917       -866   892
EMUSIC    0014       -230
END       00C0       -379  1219  1219
ENDGAM    1B30      -1130  1075
ENDRAM    4FA1      -1552  1057
ENDRND    1B2C      -1128  1210  1211
ENDSCR    4FF4       -632  1020
ERASE     190A       -861   898   900
F1        008D       -516  1776  1786  1788
F2        0046       -528
F3        0022       -540
F4        0011       -552
F5        0008       -560
FIELD     1988       -917  1084  1088
FILL      001A       -235  1024  1053  1055  1057
FIREO     17FF       -720  1217
```

```
FIRE1      180A       -724  1218
FIRST      1D6B      -1379  1238  1341  1368
FIRSTC     2000        -40
FNTSML     020D        -48   707  1060  1144
FNTSYS     0206        -47
FS1        0085       -517
FS2        0042       -529
FS3        0020       -541
FS4        0010       -553
FTBASE     0000        -93
FTBYTE     0003        -96
FTFSX      0001        -94
FTFSY      0002        -95
FTPTH      0006        -99
FTPTL      0005        -98
FTYSIZ     0004        -97
FUDG4      1F9C      -1519
FUNERL     1FC1      -1528   914
GO         00FD       -506
G1         007E       -518  1775
G2         003E       -530
G3         001F       -542
G4         000F       -554
G5         0007       -561
G6         0003       -565
G7         0001       -567
G8         0000       -568
GAMSTB     4FF8       -634  1029  1204
GETNUM     004E       -265
GETPAR     004C       -264  1018  1018
GETRDY     1D7E      -1402  1077
GFBODY     1F10      -1516  1264
GFCOLS     1DC7      -1420  1035
GFLFR      1BC5      -1174  1516
GFVSIZ     0017       -680  1296  1831  1833
GFWRIT     1B5D      -1129  1517
GFWRT1     1BA4      -1160  1243
GFWRT2     1BAC      -1161  1265  1281
GFWRT3     1BB4      -1162
GFWRT4     1BAE      -1160
GFWRT5     1BC0      -1169  1254
GOTME      1F06      -1511   938
GRX        002C       -661  1077
GRY        0001       -662  1077  1153
GSO        00EE       -507  1803
GS1        0077       -519
GS2        003B       -531
GS3        001D       -543
GS4        000E       -555
GSBEND     0007        -62  1205
GSBSCR     0001        -61  1028
GSBTIM     0000        -60
GTMINS     4FEE       -628
GTSECS     4FED       -627
GUNLMT     1D87      -1404  1394
GUNSHO     1FDB      -1530   816
GVEC3A     1CDD      -1296  1386  1391
```

```
OPOT3     4FE2        -616
OSW0      4FE4        -618
OSW1      4FE5        -619
OSW2      4FE6        -620
OSW3      4FE7        -621
PAWS      0050        -266    942    942   1142   1142   1167
PIZBRK    0048        -262    820
PJOY      1899        -812    823
POT0      001C        -201
POT1      001D        -202
POT2      001E        -203
POT3      001F        -204
PPOT      18BE        -826    838
PPOT0     18B9        -823   1212
PPOT1     18B1        -819   1213
PRIOR     4FF9        -635
PSWCY     0000         -58
PSWPV     0002         -57
PSWSGN    0007         -55
PSWZRO    0006         -56
PUTVEC    19D3        -961    804    810
PVOLAB    4FD2        -598
PVOLMC    4FD3        -599
QUIT      0078        -288   1208   1208
RANGED    0076        -287
RANSHT    4FEF        -630
RBULS     4FDB       -1554    733
RBULX     0068        -659    771   1162
RCACX     0058        -670    673    872    894    899   1032   1082
RCALL     0004        -218   1075
RCOWB     4F78       -1548    732    824    835    912   1098
RECTAN    001C        -236
RELAB1    003A        -253    780    780    882    882
RELABS    0038        -252   1321   1321
RESTOR    002E        -245
RFIELD    4FA1       -1555   1081
RFTAB     1DC2       -1420   1083
RITB      184F        -769    774
RNX       0088        -657   1068
RSCORE    4FA6       -1558    908   1072
SAVE      002C        -244
SCHEDR    000C        -224
SCREEN    0000         -41   1324   1326
SCROLL    0030        -246
SCRSTR    0016        -232
SCT0      0001        -128
SCT1      0002        -129
SCT2      0003        -130
SCT3      0004        -131
SCT4      0005        -132
SCT5      0006        -133
SCT6      0007        -134
SCT7      0008        -135   1210
SEMI4S    4FDE        -612    944   1195
SENFLG    4FFA        -636
SENTRY    0042        -259   1180
SETB      007A        -289   1028
```

```
SETOUT    0016       -233   1031
SETW      007C       -290
SF0       0009       -136   1211
SF1       000A       -137
SF2       000B       -138
SF3       000C       -139
SF4       000D       -140
SF5       000E       -141
SF6       000F       -142
SF7       0010       -143
SHIFTU    0060       -276
SINIT     1DCF      -1428   1050
SJ0       0015       -152   1214
SJ1       0017       -154   1215
SJ2       0019       -156
SJ3       001B       -158
SKYD      0013       -145   1216
SKYU      0012       -146
SNDBX     0018       -184
SNUL      0000       -127
SP0       001C       -147   1212
SP1       001D       -148   1213
SP2       001E       -149
SP3       001F       -150
SSEC      0011       -144   1219
ST0       0014       -151   1217
ST1       0016       -153   1218
ST2       0018       -155
ST3       001A       -157
STACK     4F06      -1534   1021   1024   1025
STHN      18A4       -814
STIMER    0200        -45   1427
STMRX     004C       -660    712
STOREN    0058       -272
STRDIS    0034       -249    941    941   1077   1153
STRND     1A0C      -1001   1202
STRRAM    4F12      -1537   1057   1057
STSEC     1837       -755    761
SUCK      000C       -222    725    725    731    731    904    904    911
                      911   1059   1161
SW0       0010       -197
SW1       0011       -198
SW2       0012       -199
SW3       0013       -200
SYSRAM    4FCE       -639
TAPS      1FB1      -1525    907
TBUMP     1D1E      -1325   1294   1298   1450
TBUMP1    1D25      -1330   1447
TCAC      199B       -927    957
TCACY     0014       -664    665
TIME      000B      -1430    714   1064   1070
TIMOUT    4FEC       -626
TIYU      1ABE      -1071   1125
TLINE     000A       -675    676    921   1109   1519   1525
TMR60     4FEB       -625
TONEA     0011       -177
TONEB     0012       -178
```

```
GVEC3B      1CEC    -1302   1415
GVECT       1C79    -1253   1425
GVECT1      1CB5    -1279   1377   1383
GVECT2      1CC4    -1283   1396
GVECT3      1CDA    -1294   1402
GVECT4      1CFC    -1309   1369   1442
GVECT5      1D07    -1318   1374
GVECT6      1D12    -1320   1420
HIT         18E6     -847    856
HIT1        18FB     -853    865
HIT2        1920     -874    873
HITCHK      18CE     -837   1191
HOME        1FA3    -1523   1039
HORAF       000F     -195
HORCB       0009     -173
HUMANR      0040     -257
INCSCR      0054     -268    926    926
INDEXB      005C     -274
INDEXN      0056     -271
INDEXW      005A     -273
INFBK       000D     -186   1137   1234   1344
INIT        19E8     -983    644
INITQ       1A5E    -1036
INLIN       000F     -188   1236   1350
INMOD       000E     -187
INTPC       0000     -216    706    712    725    731    780    814    830
                      870    882    904    911    926    937    941    942
                      991   1018   1023   1023   1023   1046   1046   1046
                     1115   1142   1146   1146   1146   1179   1179   1179
                     1208   1247   1252   1264   1269   1271   1321   1360
                     1396   1437
INTP@       0000     -986   -998  -1002  -1024  -1087  -1105  -1109  -1111
                    -1128
INTST       0008     -193
INTTBL      1D78    -1396   1133
JOY0        188F     -808   1214
JOY1        1895     -810   1215
KART        18CA     -834    848
KCTASC      0040     -258
KEY0        0014     -206
KEY1        0015     -207
KEY2        0016     -208
KEY3        0017     -209
KEYSEX      4FE3     -617
KIL1        1E84    -1495    917
KIL2        1ED6    -1497   1676   1698
LARG2       000C    -1432    937
LARGE       000B    -1431   1077   1153
LASTB       1FEF    -1527   1845
LBULS       4FDA    -1553    727
LBULX       0020     -658   1154
LCACX       0040     -669    896   1086   1524
LCOWB       4F61    -1546    726    822    839    905   1090   1094
LEG0        1E4F    -1489   1247   1388   1405   1470   1668
LEG1        1E64    -1491   1652
LEG2        1E74    -1493   1660
LEND        1FEF    -1560
```

```
LFIELD    4FA5      -1557  1085
LFRLIN    00C8       -686  1235
LFRVEC    1D78      -1397  1136  1233
LFTAB     1DBD      -1420  1087
LNX       0008       -655  1062
LOOP      1B07      -1109  1197  1199
LPPP2     1B19      -1116  1198
LSCORE    4FA2      -1556   915  1066
MAGIC     000C       -190   718  1311
MATH      0056       -270
MCAC      19A6       -933   963
MCACY     002A       -666  1116
MCALL     0006       -219  1159  1165
MENU      004A       -263
MENUST    0218        -50
MIDC      1AA2      -1058  1103
MJUMP     000A       -221
MOVE      005E       -275  1048
MRET      0008       -220   821  1226
MRFLOP    0006       -101   772   897   901
MRLOCK    4FF7       -633
MROR      0004       -103
MRROT     0002       -105
MRSHFT    0003       -106
MRXOR     0005       -102
MRXPND    0003       -104
MSET      1F9D      -1521  1773  1782  1792
MSKTD     007E       -291   830   830
MSTACK    4F12      -1536   814   934  1037
MUZAK     0012       -228
MUZPC     4FCE       -596
MUZSP     4FD0       -597
MXSCR     021E        -51  1018
NBRK      188D       -807  1216
NEGT      0074       -286
NEXT      FFFF       -688  1455  1477
NOGAME    0235        -53
NOPLAY    0228        -52
NORMEM    4000        -39  1053  1055  1324  1326  1817
NULPAT    1F0C      -1512  1280
NUMB      0007      -1428
NUMPLY    4FF3       -631
NWHDWR    0001        -36
OA1       008F       -576
OA2       0047       -577
OA3       0023       -578
OA4       0011       -579  1770
OA5       0008       -580
OB0       00FE       -570
OC0       00F1       -571
OD1       00D6       -572
OE1       00BF       -573
OF1       00B4       -574
OG1       00A0       -575
OPOT0     4FDF       -613
OPOT1     4FE0       -614
OPOT2     4FE1       -615
```

```
TONEC     0013         -179
TONMO     0010         -176
TOPLIN    006A         -684  1349
TREE      1DE9        -1462   976
TTREEY    000F         -665
UMARGT    4FFB         -637
UPISTR    0000         -215
USERTB    4FFD         -638
VBARM     000F         -689   690   787   850  1257  1310  1323  1413
                       1469
VBBLNK    0006          -87  1306  1313  1334
VBCCHK    0004          -84
VBCH      0003          -83
VBCL      0002          -82
VBCLAT    0003          -91   858  1360
VBCLMT    0000          -89
VBCOMP    0013         -693
VBCREV    0001          -90
VBDCH     0001          -81
VBDCL     0000          -80
VBDXH     0004          -68   832  1380
VBDXL     0003          -67   833  1379  1463
VBDYH     0009          -73   830  1382
VBDYL     0008          -72   831  1107  1381
VBLANK    0028         -242  1247  1247  1269  1269
VBLEG     0012         -692   693   917  1248  1388  1404  1407  1470
VBLEGT    0011         -691   692   916  1400  1411
VBMR      0000          -64   772   826   897   901  1095  1099  1105
                       1127  1319
VBOAH     000E          -78   689  1271  1308  1321
VBOAL     000D          -77  1272  1309  1322
VBOARM    0010         -690   691  1414  1417
VBSACT    0007          -86  1315  1332  1357  1362
VBSCHG    0003         -696  1389  1397  1410  1416  1419  1439
VBSINT    0005         -698  1253  1333  1376
VBSNOM    0004         -697  1385  1390  1398
VBSTAT    0001          -65   853   860   867   871   918  1242  1253
                       1291  1306  1313  1315  1332  1333  1334  1357
                       1362  1373  1376  1385  1389  1390  1397  1398
                       1410  1416  1419  1439  1464  1836
VBSWAG    0000         -695  1242  1373
VBTIMB    0002          -66   866  1384
VBXCHK    0007          -71   858   861  1128  1360  1465
VBXH      0006          -70   863  1108  1318  1467
VBXL      0005          -69
VBYCHK    000C          -76  1106  1129  1466
VBYH      000B          -75   879   919  1109  1317  1345  1468
VBYL      000A          -74
VECQ      4F15        -1539  1092  1096  1273  1367  1422  1437
VECSTR    4F18        -1540
VECT      003E         -255   870   870  1360  1360  1396  1396  1437
                       1437
VECTC     003C         -254
VERAF     000E         -194
VERBL     000A         -174
VIBRA     0014         -180
VOICES    4FD4         -600
```

```
VOLAB     0016        -181
VOLC      0015        -182
VOLN      0017        -183
VWRITR    001E        -237   1252   1252   1264   1264   1271   1271
WAGLMT    1D7C       -1400   1435
WAGON.    4F90       -1551    874    970   1079   1101
WAGPAT    1F40       -1518   1269
WAGVEC    4F8F       -1550   1104   1836
WAGVSZ    0012        -681   1835
WAGX      0048        -672    864
WALK      1AD5       -1085
WASTE     0FFF        -585    719   1241   1302   1326
WASTER    0FFF        -586
WINBND    0032        -683   1346
WRBL5A    1C52       -1240   1342
WRBUL1    1BE9       -1195   1338
WRBUL2    1C00       -1204   1307
WRBUL3    1C2D       -1221   1331
WRBUL4    1C31       -1223   1316
WRBUL5    1C4F       -1237   1348
WRBUL6    1C5E       -1244   1364
WRBUL7    1C70       -1248   1358   1361
WRIT      0024        -240
WRITA     0026        -241
WRITP     0022        -239    991    991   1115   1115
WRITQ     4F12       -1538   1091   1237   1340   1440
WRITR     0020        -238
WRTVEC    1D7A       -1398   1343
XINTC     0002        -217   1041   1078   1174   1184   1202
XPAND     0019        -191    952    979   1113
XPNDON    0001         -35
ZOK       1828        -745    743    748
ZORE      1813        -727    729
```